

1 WILLIAM L. ANTHONY (State Bar No. 106908)  
ERIC L. WESENBERG (State Bar No. 139696)  
2 MARK R. WEINSTEIN (State Bar No. 193043)  
ORRICK, HERRINGTON & SUTCLIFFE, LLP  
3 1000 Marsh Road  
Menlo Park, CA 94025  
4 Telephone: (650) 614-7400  
Facsimile: (650) 614-7401

5 JOHN W. KEKER (State Bar No. 49092)  
6 MICHAEL H. PAGE (State Bar No. 154913)  
KEKER & VAN NEST, LLP  
7 710 Sansome Street  
San Francisco, CA 94111-1704  
8 Telephone: (415) 391-5400  
Facsimile: (415) 397-7188

9 Additional Counsel Listed at Signature Block

10  
11 UNITED STATES DISTRICT COURT  
12 NORTHERN DISTRICT OF CALIFORNIA  
13 OAKLAND DIVISION  
14

15 INTERTRUST TECHNOLOGIES  
CORPORATION, a Delaware corporation,

16 Plaintiff,

17 v.

18 MICROSOFT CORPORATION, a  
19 Washington corporation,

20 Defendant.

21  
22 MICROSOFT CORPORATION, a  
Washington corporation,

23 Counterclaimant,

24 v.

25 INTERTRUST TECHNOLOGIES  
CORPORATION, a Delaware corporation,

26 Counter Claim-Defendant.  
27  
28

**BEST AVAILABLE COPY**

Case No. C 01-1640 SBA (MJE)

**PATENT LOCAL RULE 4-3 JOINT  
CLAIM CONSTRUCTION AND  
PREHEARING STATEMENT  
REVISED IN ACCORDANCE WITH  
THE SCOPE OF "MINI-MARKMAN"  
HEARING SET FORTH IN THE  
COURT'S ORDER ENTERED 2/24/03**

1 In accordance with the Court's Order entered February 24, 2003 and Patent Local  
2 Rule 4-3, Plaintiff and Counter-Defendant InterTrust Technologies ("InterTrust") and Defendant  
3 and Counter-Claimant Microsoft Corporation ("Microsoft") submit the following revised Joint  
4 Claim Construction and Prehearing Statement. Pursuant to that Order, the parties have limited  
5 their disputes for purposes of the "Mini-Markman" proceeding, to 30 disputed terms and phrases,  
6 as identified in alphabetical order in Exhibit B and highlighted in copies of the claims in Exhibit  
7 H, hereto.

#### 8 Submission of "Intrinsic" Evidence

9 To avoid unnecessary duplication, the parties will submit, prior to the submission  
10 of the final briefs in the "Mini-Markman" proceeding (including briefing addressing  
11 indefiniteness), a Joint Declaration presenting the Intrinsic evidence (including patents, file  
12 histories and cited references). The parties agree that in briefs submitted in the "Mini-Markman"  
13 proceeding, a party may cite to evidence that ultimately will be submitted by the parties in such  
14 Joint Declaration and need not append such evidence to a declaration in support of a brief. This  
15 agreement does not limit either party from submitting any evidence with a declaration  
16 accompanying any brief.

#### 17 RULE 4-3(a): Agreed Construction

- 18 • Attached hereto as **Exhibit I** is a list of claim constructions upon which the parties agree.  
19 To the extent that agreed constructions refer to disputed terms that are not among the 30  
20 terms in the "Mini-Markman" proceeding, such terms are set forth in quotations.

#### 21 RULE 4-3(b): Disputed Claim Construction Presentation

- 22 • Attached hereto as **Exhibit A** is a list of disputed claim terms set forth in claim order,  
23 together with the parties' proposed constructions.  
24 • Attached hereto as **Exhibit B** is a list of the 30 disputed claim terms in alphabetical order,  
25 together with the parties' proposed constructions.  
26 • Attached hereto as **Exhibit C** is InterTrust's identification of intrinsic and extrinsic  
27 evidence supporting its proposed construction for each of the 30 disputed terms and  
28 phrases.

- 1 • Attached hereto as **Exhibit D** is Microsoft's identification of intrinsic and extrinsic  
2 evidence supporting its proposed construction for each of the 30 disputed terms and  
3 phrases.  
4 • Attached hereto as **Exhibit E** is a Microsoft statement of reservations.  
5 • Attached hereto as **Exhibit H** is the text of the 12 claims at issue, with bolding identifying  
6 the terms and phrases in dispute for the purposes of the "Mini-Markman" proceeding.

7 **RULE 4-3(c): Claim Construction Hearing Length**

8 The claim construction schedule is set forth in the Court's Order entered February  
9 24, 2003.

10 **RULE 4-3(d): Witness Testimony**

11 The parties have agreed to present witness testimony through declarations filed in  
12 support of the briefs. There also shall be tutorial presentations, per the Court's Order of February  
13 24, 2003.

- 14 • Attached hereto as **Exhibit F** is a summary of expert testimony to be presented by  
15 InterTrust.  
16 • Attached hereto as **Exhibit G** is a summary of expert testimony to be presented by  
17 Microsoft.

18 **RULE 4-3(e): Pre-Hearing Conference Issues**

19 The parties addressed pre-hearing matters at the Case Management Conference  
20 hearing on February 13, 2003. No pre-hearing conference is currently scheduled or requested.

21 Dated: March 14, 2003

22 INTERTRUST TECHNOLOGIES  
23 CORPORATION  
24 MARK SCADINA - #173103  
25 JEFF MCDOW - #184727  
26 4800 Patrick Henry Drive  
27 Santa Clara, CA 95054  
28 Telephone: (408) 855-0100  
Facsimile: (408) 855-0144

By: 

Jeff McDow

FOR JEFF McDOW

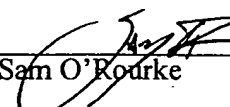
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

KEKER & VAN NEST, LLP  
MICHAEL H. PAGE

DERWIN & SIEGEL  
DOUGLAS K. DERWIN - #111407  
3280 Alpine Road  
Portola Valley, CA 94028  
Telephone: (650) 529-8700  
Facsimile: (650) 529-8799

Attorneys for Plaintiff and Counter-Defendant  
INTERTRUST TECHNOLOGIES  
CORPORATION  
WILLIAM L. ANTHONY  
HEIDI L. KEEFE  
MARK R. WEINSTEIN  
ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: March 14, 2003

By:   
Sam O'Rourke

KLARQUIST SPARKMAN, LLP  
One World Trade Center  
121 S.W. Salmon, Suite 1600  
Portland, OR 97204  
Telephone: (503) 226-7391  
Facsimile: (503) 228-9446

Attorneys for Microsoft Corporation



# EXHIBIT A

## Intertrust v. MS: JCCS Claim Chart

### U.S. Patent No. 6,253,193, Asserted Claim 1

|    | <u>'193 Claim 1</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|----|---|---|---|
| 1. | 1. A method comprising:   | The claim contains no requirement of a VDE.   | Claim as a whole: The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.)   |
| 2. | receiving a digital file including music,                             |   |   |
| 3. | storing said digital file in a first secure memory of a first device; | <u>secure</u> : One or more mechanisms are employed to prevent, detect or discourage misuse of or interference with information or processes. Such mechanisms may include concealment, Tamper Resistance, Authentication and access control. Concealment means that it is difficult to read information (for example, programs may be encrypted). Tamper Resistance and Authentication are separately defined (see item #67 and item #27, respectively, below). Access control means that access to information or processes is limited on the basis of authorization. Security is not absolute, but is designed to be sufficient for a particular purpose. | <u>secure</u> : (1) A state in which all users of a system are guaranteed that all information, processes, and devices within the system, shall have their availability, secrecy, integrity, authenticity and nonrepudiation maintained against all of the identified threats thereto.<br>(2) "Availability" means the property that information is accessible and usable upon demand by authorized persons, at least to the extent that no user may delete the information without authorization.<br>(3) "Secrecy," also referred to as confidentiality, means the property that information (including computer processes) is not made available or disclosed to unauthorized persons or processes.<br>(4) "Integrity" means the property that information has not been altered either intentionally or accidentally.<br>(5) "Authenticity" means the property that the characteristics asserted about a person, device, program, information, or process are genuine and timely, particularly as to identity, data integrity, and origin integrity.<br>(6) "Nonrepudiation" means the property that a sender of information cannot deny its origination and that a recipient of information cannot deny its receipt. |

|    | <u>'193 Claim 1</u>  | <u>IT Construction</u>   | <u>MS Construction</u>  |
|----|--|--|---|
| 4. | storing information associated with said digital file in a <b>secure</b> database stored on said first device, said information including at least one <b>budget control</b> and | <p><u>secure</u>: see item #3 above</p> <p><u>budget</u>: Information specifying a limitation on usage.</p> <p><u>control</u>: Information and/or programming controlling operations on or use of resources (e.g., content) including (a) permitted, required or prevented operations, (b) the nature or extent of such operations or (c) the consequences of such operations.</p> | <p><u>secure</u>: see item #3 above</p> <p><u>budget</u>: (1) A unique type of "method" that specifies a decrementable numerical limitation on future Use (e.g., copying) of digital information and how such Use will be paid for, if at all.<br/>(2) A "method" is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements, and/or relationships for use in performing, and/or preparing to perform, basic instructions in relation to the operation of one or more electronic appliances.</p> <p><u>control</u>: (1) Independent, special-purpose, <b>Executable</b>, which can execute only within a <i>Secure Processing Environment</i> (see below).<br/>(2) Each <b>VDE Control</b> is a <b>Component Assembly</b> dedicated to a particular activity (e.g., editing, modifying another <b>Control</b>, a user-defined action, etc.), particular user(s), and particular protected information, and whose satisfactory execution is necessary to <i>Allowing</i> (see below) that activity.<br/>(3) Each separate information <i>Access</i> (see below) or <i>Use</i> is independently <b>Controlled</b> by independent <b>VDE Control(s)</b>.<br/>(4) Each <b>VDE Control</b> is assembled within a <i>Secure Processing Environment</i> from independently deliverable modular components (e.g., <i>Load Modules</i> (see below) or other <b>Controls</b>), dynamically in response to an information <i>Access</i> or <i>Use Request</i>.<br/>(5) The dynamic assembly of a <b>Control</b> is directed by a "blueprint" <i>Record</i> (see below) (put in place by one or more <b>VDE users</b>) <b>Containing</b> control information identifying the exact modular code components to be</p> |

|  | <u>'193 Claim 1</u> | <u>IT Construction</u> | <u>MS Construction</u>   |
|--|---------------------|------------------------|--|
|  |                     |                        | <p>assembled and executed to govern (i.e., <b>Control</b>) this particular activity on this particular information by this particular user(s).</p> <p>(6) Each <b>Control</b> is independently assembled, loaded and delivered vis-à-vis other <b>Controls</b>.</p> <p>(7) Control information and <b>Controls</b> are extensible and can be configured and modified by all users, and combined by all users with any other <b>VDE</b> control information or <b>Controls</b> (including that provided by other users), subject only to "senior" user <b>Controls</b>.</p> <p>(8) Users can assign control information (including alternative control information) and <b>Controls</b> to an arbitrarily fine, user-defined portion of the protected information, such as a single paragraph of a document, as opposed to being limited to file-based controls.</p> <p>(9) <b>VDE Controls</b> reliably limit Use of the protected information to only authorized activities and amounts.</p> <p>For the purposes of the construction of "Control," a "<i>Secure Processing Environment</i>" is defined as: A Secure Processing Environment is uniquely identifiable, self-contained, non-circumventable, and trusted by all other <b>VDE</b> nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the patent application as being protected, and to guarantee that such information will be accessed and <b>Used</b> only as expressly authorized by the associated <b>VDE Controls</b>, and to guarantee that all requested reporting of and payments for protected information use will be made. A Secure Processing Environment is formed by, and requires, a Secure Processing Unit having a hardware <b>Tamper Resistant Barrier</b> encapsulating a processor and internal</p> |

|  | <u>'193 Claim 1</u> | <u>IT Construction</u> | <u>MS Construction</u>   |
|--|---------------------|------------------------|--|
|  |                     |                        | <p>Secure memory. The <b>Tamper Resistant Barrier</b> prevents all unauthorized interference, removal, observation, and other Use of the information and processes within it.</p> <p>For the purposes of the construction of "Control," "<i>Allowing</i>" is defined as: Actively permitting an action that otherwise cannot be taken (i.e., is prohibited) by any user, process, or device. In <b>VDE</b>, an action is allowed only through execution (within a <i>Secure Processing Environment</i>) of the <b>VDE Control(s)</b> assigned to the particular action request, and satisfaction of all requirements imposed by such execution.</p> <p>For the purposes of the construction of "Control," "<i>Access</i>" is defined as: To satisfactorily perform the steps necessary to obtain something so that it can be <b>Used</b> in some manner (e.g., for information: copied, printed, decrypted, encrypted, saved, modified, observed, or moved, etc.). In <b>VDE</b>, access to protected information is achieved only through execution (within a <i>Secure Processing Environment</i>) of the <b>VDE Control(s)</b> assigned to the particular "access" request, satisfaction of all requirements imposed by such execution, and the <b>Controlled opening of the Secure Container Containing</b> the information.</p> <p>For the purposes of the construction of "Control," a "<i>Load Module</i>" is defined as: An <b>Executable</b>, modular unit of machine code (which may include data) suitable for loading into memory for execution by a processor. A load module is encrypted (when not within a secure processing unit) and has an <b>Identifier</b> that a calling process must provide to be able to use the load module. A load module is combinable with other load modules,</p> |

|    | <u>'193 Claim 1</u>                | <u>IT Construction</u>  | <u>MS Construction</u>   |
|----|------------------------------------|---|--|
|    |                                    |   | <p>and associated data, to form <b>Executable Component Assemblies</b>. A load module can execute only in a <b>VDE Protected Processing Environment</b>. Library routines are not load modules and dynamic link libraries are not load modules.</p> <p>For the purposes of the construction of "Control," a "<i>Record</i>" is defined as: A data structure that is a collection of fields (elements), each with its own name and type. Unlike an array, whose elements are accessed using an index, the elements of a record are accessed by name. A record can be accessed as a collective unit of elements, or the elements can be accessed individually.</p>   |
| 5. | at least one <b>copy control</b> , | <p><u>copy</u>: To reproduce. The reproduction must be usable, may incorporate all of the original item or only some of it, and may involve some changes to the item as long as the essential nature of the content remains unchanged.</p> <p><u>control</u>: see item #4 above</p> | <p><u>copy</u>: (1) To reproduce all of a <i>Digital File</i> (see below) or other complete physical block of data from one location on a storage medium to another location on the same or different storage medium, leaving the original block of data unchanged, such that two distinct and independent objects exist.</p> <p>(2) Although the layout of the data values in physical storage may differ from the original, the resulting "copy" is logically indistinguishable from the original.</p> <p>(3) The resulting "copy" may or may not be encrypted, ephemeral, usable, or accessible.</p> <p>For the purposes of the construction of "Copy," a "<i>Digital File</i>" is defined as: A named, static unit of storage allocated by a "file system" and <b>Containing</b> digital information. A digital file enables any application using the "file system" to randomly access its contents and to distinguish it by name from every other such unit. A copy of a digital file is a separate digital file. A "file system" is the portion of the operating system</p> |

**EXHIBIT A TO JOINT CLAIM CONSTRUCTION STATEMENT**

|    | <u>'193 Claim 1</u>  | <u>IT Construction</u>   | <u>MS Construction</u>   |
|----|--|--|--|
|    |  |  | that translates requests made by application programs for operations on "files" into low-level tasks that can control storage devices such as disk drives.<br><br><u>control</u> : see item #4 above   |
| 6. | said at least one <b>budget control</b> including <i>a budget specifying the number of copies which can be made of said digital file</i> ; | <u>budget</u> : see item #4 above<br><br><u>control</u> : see item #4 above<br><br><u>a budget specifying the number of copies which can be made of said digital file</u> : Normal English, incorporating the separately defined terms: a Budget stating the number of copies that can be made of the digital file referred to earlier in the claim.   | <u>budget</u> : see item #4 above<br><br><u>control</u> : see item #4 above<br><br><u>a budget specifying the number of copies which can be made of said digital file</u> : A <b>Budget</b> explicitly stating the total number of copies (whether or not decrypted, long-lived, or accessible) that (since creation of the <b>Budget</b> ) are authorized to be made of the <i>Digital File</i> by any and all users, devices, and processes. No process, user, or device is able to make another copy of the <i>Digital File</i> once this number of copies has been made.<br><br>For the purposes of the construction of this phrase, " <i>Digital File</i> " is defined as set forth in item #5, above.                                  |
| 7. | and said at least one <b>copy control</b> <i>controlling the copies made of said digital file</i> ;  | <u>copy</u> : see item #5 above<br><br><u>control</u> : see item #4 above<br><br><u>controlling</u> : Normal English: exercising authoritative or dominating influence over; directing.<br><br><u>controlling the copies made of said digital file</u> : The nature of this operation is further defined in later claim elements. In context, the copy control determines the conditions under which a digital file may be Copied and the copied file stored on a second device. | <u>copy</u> : see item #5 above<br><br><u>control</u> : see item #4 above<br><br><u>controlling</u> : (1) Reliably defining and enforcing the conditions and requirements under which an action that otherwise cannot be taken, will be <i>Allowed</i> , and the manner in which it may occur. Absent verified satisfaction of those conditions and requirements, the action cannot be taken by any user, process or device. (2) In VDE, an action is <b>Controlled</b> through execution of the applicable <b>VDE Control(s)</b> within a <b>VDE Secure Processing Environment</b> . (3) More specifically, in VDE, <b>Controlling</b> is effected by use of <b>VDE Controls</b> , <b>VDE Secure Containers</b> , and <b>VDE foundation</b> |

EXHIBIT A TO JOINT CLAIM CONSTRUCTION STATEMENT

| <u>'193 Claim 1</u> | <u>IT Construction</u> | <u>MS Construction</u>  |
|---------------------|------------------------|---|
|                     |                        | <p>(including <i>VDE Secure Processing Environment</i>, "object registration," and other mechanisms for allegedly individually ensuring that specific <b>Controls</b> are enforced vis-à-vis specific objects (and their content at an arbitrary granular level) and specific "users").</p> <p>For the purposes of the construction of "Control (v.)" et al, "<i>Allowed</i>" and "<i>Secure Processing Environment</i>" are defined as set forth in item #4, above.</p> <p><u>controlling the copies made of said digital file</u>: <b>Controlling Uses</b> of and <i>Accesses</i> to all copies of the <i>Digital File</i>, by all users, processes, and devices, by executing each of the recited "at least one" <b>Copy Control(s)</b> within <i>VDE Secure Processing Environment(s)</i>. Each <b>Control</b> governs (<b>Controls</b>) only one action, which action may or may not differ among the different "at least one" <b>Controls</b>. All <i>Uses</i> and <i>Accesses</i> are prohibited and incapable of occurring except to the extent <i>Allowed</i> by the "at least one" <b>Copy Control(s)</b>.</p> <p>For the purposes of the construction of this phrase, "<i>Secure Processing Environment</i>," "<i>Access</i>" and "<i>Allowed</i>" are defined as set forth in item #4, above.</p> |

|     | <u>'193 Claim 1</u>   | <u>IT Construction</u>   | <u>MS Construction</u>   |
|-----|---|--|--|
| 8.  | determining whether said digital file may be <b>copied</b> and stored on a second device based on at least said <b>copy control</b> ; | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above |
| 9.  | if said <b>copy control</b> allows at least a portion of said digital file to be <b>copied</b> and stored on a second device,         | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above |
| 10. | <b>copying</b> at least a portion of said digital file;   | <u>copying (copy)</u> : see item #5 above                                      | <u>copying (copy)</u> : see item #5 above                                      |
| 11. | transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;          |  |  |
| 12. | storing said digital file in said memory of said second device; and   |  |  |
| 13. | including playing said music through said audio output.   |  |  |



**U.S. Patent No. 6,253,193, Asserted Claim 11**

|     | <b><u>'193 Claim 11</u></b>  | <b><u>IT Construction</u></b>  | <b><u>MS Construction</u></b>   |
|-----|--|--|---|
| 14. | 11. A method comprising:   | The claim contains no requirement of a VDE.                                    | <u>Claim as a whole</u> : The recited method is performed within a <b>VDE</b> . (See item #86 for Microsoft's construction of <b>VDE</b> .) |
| 15. | receiving a digital file;  |  |   |
| 16. | storing said digital file in a first <b>secure</b> memory of a first device;   | <u>secure</u> : see item #3 above  | <u>secure</u> : see item #3 above   |
| 17. | storing information associated with said digital file in a <b>secure</b> database stored on said first device, said information including a first <b>control</b> ;   | <u>secure</u> : see item #3 above<br><u>control</u> : see item #4 above        | <u>secure</u> : see item #3 above<br><u>control</u> : see item #4 above   |
| 18. | determining whether said digital file may be <b>copied</b> and stored on a second device based on said first <b>control</b> , said determining step including identifying said second device and determining whether,              | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above | <u>copied (copy)</u> : see item #5 above<br><u>control</u> : see item #4 above  |
| 19. | said first <b>control</b> allows transfer of said <b>copied</b> file to said second device, said determination based at least in part on the features present at the device to which said <b>copied</b> file is to be transferred; | <u>control</u> : see item #4 above<br><u>copied (copy)</u> : see item #5 above | <u>control</u> : see item #4 above<br><u>copied (copy)</u> : see item #5 above  |

|     | <u>'193 Claim 11</u>   | <u>IT Construction</u>   | <u>MS Construction</u>   |
|-----|--|--|--|
| 20. | if said first <b>control</b> allows at least a portion of said digital file to be <b>copied</b> and stored on a second device, | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above |
| 21. | <b>copying</b> at least a portion of said digital file;  | <u>copying (copy)</u> : see item #5 above  | <u>copying (copy)</u> : see item #5 above  |
| 22. | transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;   |  |  |
| 23. | storing said digital file in said memory of said second device; and  |  |  |
| 24. | rendering said digital file through said output.   |  |  |

|     | <u>'193 Claim 15</u>   | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|--|--|---|
| 25. | 15. A method comprising:   | The claim contains no requirement of a VDE.  | <u>Claim as a whole:</u> The recited method is performed within a VDE. (See item #93 for Microsoft's construction of VDE.)  |
| 26. | receiving a digital file;  |  |   |
| 27. | an <b>authentication</b> step comprising:  | <u>authentication:</u> Identifying (e.g., a person, device, organization, document, file, etc.). Includes uniquely identifying or identifying as a member of a group.  | <u>authentication:</u> To establish that the following asserted characteristics of something (e.g., a person, device, organization, document, file, etc.) are genuine: its identity, its data integrity, (i.e., it has not been altered) and its origin integrity (i.e., its source and time of origination). |
| 28. | accessing at least one <b>identifier</b> associated with a first device or with a user of said first device; and   | <u>identifier:</u> Information used to identify something or someone (e.g., a password).<br><br>In this definition, "identify" means to establish the identity of or to ascertain the origin, nature, or definitive characteristics of; includes identifying as an individual or as a member of a group. | <u>identifier:</u> Any text string used as a label naming an individual instance of what it <i>Identifies</i> (see below)<br><br>For the purpose of the construction of "Identifier," "Identify" is defined as: To establish as being a particular instance of a person or thing.                             |
| 29. | determining whether said <b>identifier</b> is associated with a device and/or user authorized to store said digital file;  | <u>identifier:</u> see item #28 above  | <u>identifier:</u> see item #28 above   |
| 30. | storing said digital file in a first <b>secure</b> memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized; | <u>secure:</u> see item #3 above   | <u>secure:</u> see item #3 above  |
| 31. | storing information associated with said digital file in a <b>secure</b> database stored on said first   | <u>secure:</u> see item #3 above<br><br><u>control:</u> see item #4 above  | <u>secure:</u> see item #3 above<br><br><u>control:</u> see item #4 above   |

|     | <u><b>'193 Claim 15</b></u>   | <u><b>IT Construction</b></u>  | <u><b>MS Construction</b></u>  |
|-----|---|--|--|
|     | device, said information including at least one <b>control</b> ;  |  |  |
| 32. | determining whether said digital file may be <b>copied</b> and stored on a second device based on said at least one <b>control</b> ;  | <u>copied (copy)</u> : see item #5 above<br><br><u>control</u> : see item #4 above | <u>copied (copy)</u> : see item #5 above<br><br><u>control</u> : see item #4 above |
| 33. | if said at least one <b>control</b> allows at least a portion of said digital file to be <b>copied</b> and stored on a second device, | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above |
| 34. | <b>copying</b> at least a portion of said digital file;   | <u>copying (copy)</u> : see item #5 above  | <u>copying (copy)</u> : see item #5 above  |
| 35. | transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;          |  |  |
| 36. | storing said digital file in said memory of said second device; and   |  |  |
| 37. | rendering said digital file through said output.  |  |  |

|     | <u>'193 Claim 19</u>   | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|--|--|---|
| 38. | 19. A method comprising:   | The claim contains no requirement of a VDE.  | <u>Claim as a whole:</u> The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.)  |
| 39. | receiving a digital file at a first device;  |  |   |
| 40. | establishing communication between said first device and a <b>clearinghouse</b> located at a location remote from said first device;   | <u>clearinghouse:</u> A provider of financial and/or administrative services for a number of entities; or an entity responsible for the collection, maintenance, and/or distribution of materials, information, licenses, etc. | <u>clearinghouse:</u> (1) A computer system that provides intermediate storing and forwarding services for both content and audit information, and which two or more parties trust to provide its services independently because it is operated under constraint of VDE security. (2) "Audit information" means all information created, stored, or reported in connection with an "auditing" process. "Auditing" means tracking, metering and reporting the usage of particular information or a particular appliance. |
| 41. | said first device obtaining authorization information including a key from said <b>clearinghouse</b> ;   | <u>clearinghouse:</u> see item #40 above   | <u>clearinghouse:</u> see item #40 above  |
| 42. | said first device using said authorization information to gain access to or make at least one use of said first digital file, including using said key to decrypt at least a portion of said first digital file; and | <u>use:</u> Normal English: to put into service or apply for a purpose, to employ.   | <u>use:</u> (1) To use information is to perform some action on it or with it (e.g., copying, printing, decrypting, encrypting, saving, modifying, observing, or moving, etc.). (2) In VDE, information Use is <i>Allowed</i> only through execution of the applicable VDE Control(s) and satisfaction of all requirements imposed by such execution.<br><br>For the purposes of the construction of "Use," "Allowed" is defined as set forth in item #4, above.  |
| 43. | receiving a first <b>control</b> from said <b>clearinghouse</b> at said first device;  | <u>control:</u> see item #4 above<br><br><u>clearinghouse:</u> see item #40 above  | <u>control:</u> see item #4 above<br><br><u>clearinghouse:</u> see item #40 above   |

|     | <u>'193 Claim 19</u>   | <u>IT Construction</u>   | <u>MS Construction</u>   |
|-----|--|--|--|
| 44. | storing said first digital file in a memory of said first device;  |  |  |
| 45. | using said first <b>control</b> to determine whether said first digital file may be <b>copied</b> and stored on a second device;     | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above |
| 46. | if said first <b>control</b> allows at least a portion of said first digital file to be <b>copied</b> and stored on a second device, | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above | <u>control</u> : see item #4 above<br><br><u>copied (copy)</u> : see item #5 above |
| 47. | <b>copying</b> at least a portion of said first digital file;  | <u>copying (copy)</u> : see item #5 above  | <u>copying (copy)</u> : see item #5 above  |
| 48. | transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;   |  |  |
| 49. | storing said first digital file portion in said memory of said second device; and  |  |  |
| 50. | rendering said first digital file portion through said output.   |  |  |

|     | <u>'683 Claim 2</u>          | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|------------------------------|--|---|
| 51. | 2. A system including:       | The claim contains no requirement of a VDE.  | Claim as a Whole: The "system" is a VDE. (See item #86 for Microsoft's construction of VDE.)  |
| 52. | a first apparatus including, |  |   |
| 53. | user controls,               | <u>control</u> : see item #4 above   | <u>control</u> : see item #4 above  |
| 54. | a communications port,       |  |   |
| 55. | a processor,                 |  |   |
| 56. | a memory storing:            |  |   |
| 57. | a first secure container     | <p><u>secure container</u>: A container that is Secure.</p> <p>In this definition, "container" means a digital file containing linked and/or embedded items.</p> | <p><u>secure container</u>: (1) A <b>VDE Secure Container</b> is a self-contained, self-protecting data structure which (a) encapsulates information of arbitrary size, type, format, and organization, including other, nested, containers, (b) cryptographically protects that information from all unauthorized <i>Access</i> and <i>Use</i>, (c) provides encrypted storage management functions for that information, such as hiding the physical storage location(s) of its protected contents, (d) permits the association of itself or its contents with <b>Controls</b> and control information governing (<b>Controlling</b>) <i>Access</i> to and <i>Use</i> thereof, and (e) prevents such <i>Use</i> or <i>Access</i> (as opposed to merely preventing decryption) until it is "opened."</p> <p>(2) A <b>Secure Container</b> can be opened only as expressly <i>Allowed</i> by the associated <b>VDE Control(s)</b>, only within a <i>Secure Processing Environment</i>, and only through decryption of its encrypted header.</p> <p>(3) A <b>Secure Container</b> is not directly accessible to any non-VDE or user calling process. All such calls are intercepted by VDE.</p> <p>(4) The creator of a <b>Secure Container</b> can assign (or allow others to assign) control information to any arbitrary portion of a <b>Secure Container's</b> contents, or to an empty <b>Secure Container</b> (to govern</p> |

|     | <u>'683 Claim 2</u>   | <u>IT Construction</u>  | <u>MS Construction</u>   |
|-----|---|---|--|
|     |   |   | <p>(Control) the later addition of contents to the container, and Access to or Use of those contents).</p> <p>(5) A container is not a <b>Secure Container</b> merely because its contents are encrypted and signed. A <b>Secure Container</b> is itself <b>Secure</b>.</p> <p>(6) All VDE-protected information (including protected content, information about content usage, content-control information, <b>Controls</b>, and <i>Load Modules</i>) is encapsulated within a <b>Secure Container</b> whenever stored outside a <i>Secure Processing Environment</i> or secure database.</p> <p>For the purposes of the construction of "Secure Container," "<i>Secure Processing Environment</i>," "<i>Load Module</i>," "<i>Access</i>" and "<i>Allow</i>" are defined as set forth in item #4, above.</p> |
| 58. | containing a governed item,   | <u>containing</u> : Normal English: having within or holding. In the context of an element contained within a data structure (e.g., a secure container), the contained element may be either directly within the container or the container may hold a reference indicating where the element may be found. | <u>containing</u> : Physically (directly) storing within, as opposed to addressing (i.e., referring to something by the explicitly identified location where it is stored, without directly storing it).   |
| 59. | the first secure container governed item being at least in part encrypted; the first secure container having been received from a second apparatus; | <u>secure container</u> : see item #57 above  | <u>secure container</u> : see item #57 above   |



|     | <u>'683 Claim 2</u>   | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|---|--|---|
| 60. | a first <b>secure container</b> rule at least in part governing an <b>aspect</b> of access to or <b>use</b> of said first <b>secure container</b> governed item, the first <b>secure container</b> rule, the first <b>secure container</b> rule having been received from a third apparatus different from said second apparatus; and | <p><u>secure container</u>: see item #57 above</p> <p><u>aspect</u>: Feature, element, property or state.</p> <p><u>use</u>: see item #42 above</p>  | <p><u>secure container</u>: see item #57 above</p> <p><u>aspect</u>: An aspect of an environment is a persistent element or property of that environment that can be used to distinguish it from other environments.</p> <p><u>use</u>: see item #42 above</p>  |
| 61. | hardware or software used for receiving and opening <b>secure containers</b> , said <b>secure containers</b> each including the capacity to <b>contain</b> a governed item, a <b>secure container</b> rule being associated with each of said <b>secure containers</b> ;  | <p><u>secure container</u>: see item #57 above</p> <p><u>contain (containing)</u>: see item #58 above</p>  | <p><u>secure container</u>: see item #57 above</p> <p><u>contain (containing)</u>: see item #58 above</p>   |
| 62. | a <b>protected processing environment</b> at least in part protecting information <b>contained</b> in said <b>protected processing environment</b> from tampering by a user of said first apparatus,  | <p><u>protected processing environment</u>: An environment in which processing and/or data is at least in part protected from tampering. The level of protection can vary, depending on the threat.</p> <p>In this definition, "environment" means capabilities available to a program running on a computer or other device or to the user of a computer or other device. Depending on the context, the environment may be in a single device (e.g., a personal computer) or may be spread among multiple</p> | <p><u>protected processing environment</u>:<br/> (1) A uniquely identifiable, self-contained computing base trusted by all <b>VDE</b> nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the February, 1995, patent application as being protected, and to guarantee that such information will be <i>Accessed</i> and <i>Used</i> only as expressly authorized by <b>VDE Controls</b>.<br/> (2) At most <b>VDE</b> nodes, the <b>Protected Processing Environment</b> is a <i>Secure Processing Environment</i> which is formed by, and requires, a</p> |

|  | <u>'683 Claim 2</u> | <u>IT Construction</u>   | <u>MS Construction</u>   |
|--|---------------------|--|--|
|  |                     | <p>devices (e.g., a network).</p> <p><u>contained (containing):</u> see item #58 above</p> | <p>hardware <b>Tamper Resistant Barrier</b> encapsulating a special-purpose Secure Processing Unit having a processor and internal secure memory. "Encapsulated" means hidden within an object so that it is not directly accessible but rather is accessible only through the object's restrictive interface.</p> <p>(3) The <b>Tamper Resistant Barrier</b> prevents all unauthorized (intentional or accidental) interference, removal, observation, and use of the information and processes within it, by all parties (including all users of the device in which the <b>Protected Processing Environment</b> resides), except as expressly authorized by <b>VDE Controls</b>.</p> <p>(4) A <b>Protected Processing Environment</b> is under <b>Control of Controls</b> and control information provided by one or more parties, rather than being under <b>Control</b> of the appliance's users or programs.</p> <p>(5) Where a <b>VDE</b> node is an established financial <b>Clearinghouse</b>, or other such facility employing physical facility and user-identity <b>Authentication</b> security procedures trusted by all <b>VDE</b> nodes, and the <b>VDE</b> node does not <b>Access</b> or <b>Use</b> <b>VDE</b>-protected information, or assign <b>VDE</b> control information, then the <b>Protected Processing Environment</b> at that <b>VDE</b> node may instead be formed by a general-purpose CPU that executes all <b>VDE</b> "security" processes in protected (privileged) mode.</p> <p>(6) A <b>Protected Processing Environment</b> requires more than just verifying the integrity of <b>Digitally Signed Executable</b> programming prior to execution of the programming; or concealment of the program, associated data, and execution of the program code; or use of a password as its protection</p> |

|     | <u>'683 Claim 2</u>  | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|--|---|---|
|     |  |   | <p>mechanism.</p> <p>For the purposes of the construction of "Protected Processing Environment," "<i>Secure Processing Environment</i>" and "Access" are defined as set forth in item #4, above.</p> <p><u>contained (containing)</u>: see item #58 above</p> |
| 63. | <p>said <b>protected processing environment</b> including hardware or software used for applying said first <b>secure container</b> rule and a second <b>secure container</b> rule in combination to at least in part govern at least one <b>aspect</b> of access to or <b>use</b> of a governed item <b>contained</b> in a <b>secure container</b>; and</p> | <p><u>protected processing environment</u>: see item #62 above</p> <p><u>secure container</u>: see item #57 above</p> <p><u>aspect</u>: see item #60 above</p> <p><u>use</u>: see item #42 above</p> <p><u>contained (containing)</u>: see item #58 above</p> | <p><u>protected processing environment</u>: see item #62 above</p> <p><u>secure container</u>: see item #57 above</p> <p><u>aspect</u>: see item #60 above</p> <p><u>use</u>: see item #42 above</p> <p><u>contained (containing)</u>: see item #58 above</p> |
| 64. | <p>hardware or software used for transmission of <b>secure containers</b> to other apparatuses or for the receipt of <b>secure containers</b> from other apparatuses.</p>  | <p><u>secure container</u>: see item #57 above</p>  | <p><u>secure container</u>: see item #57 above</p>  |

|     | <u>'721 Claim 1</u>   | <u>IT Construction</u>  | <u>MS Construction</u>   |
|-----|---|---|--|
| 65. | 1. A security method comprising:  | The claim contains no requirement of a VDE.   | <u>Claim as a whole</u> : The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.)  |
| 66. | <b>digitally signing</b> a first load module with a first <b>digital signature</b> designating the first load module for use by a first <b>device class</b> ; | <p><u>digital signature</u>: A digital value, verifiable with a key, that can be used to determine the source and/or integrity of a signed item (e.g., a file, program, etc.).</p> <p>Digitally signing is the process of creating a digital signature.</p> <p><u>designating</u>: Normal English: indicating, specifying, pointing out or characterizing.</p> <p><u>use</u>: see item #42 above</p> <p><u>device class</u>: A group of devices which share at least one attribute.</p> | <p><u>digitally signing</u>:</p> <p>(1) Creating a <b>Digital Signature</b> using a secret <i>Key</i> (see below).</p> <p>(2) In symmetric key cryptography, a "secret key" is a <i>Key</i> that is known only to the sender and recipient. In asymmetric key cryptography, a "secret key" is the private <i>Key</i> of a public/private key pair, in which the two keys are related uniquely by a predetermined mathematical relationship such that it is computationally infeasible to determine one from the other.</p> <p>For the purposes of the construction of "Digital Signing," a "<i>Key</i>" is defined as: A bit sequence used and needed by a cryptographic algorithm to encrypt a block of plain text or to decrypt a block of cipher text. A key is different from a key seed or other information from which the actual encryption and/or decryption key is constructed, <b>Derived</b>, or otherwise identified. In symmetric key cryptography, the same key is used for both encryption and decryption. In asymmetric or "public key" cryptography, two related keys are used; a block of text encrypted by one of the two keys (e.g., the "public key") can be decrypted only by the corresponding key (e.g., the "private key").</p> <p><u>digital signature</u>: A computationally unforgeable string of characters (e.g., bits) generated by a cryptographic operation on a block of data using some secret. The string can be generated only by an entity that knows the secret, and hence provides</p> |

|     | <u>'721 Claim 1</u>  | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|--|--|---|
|     |  |  | <p>evidence that the entity must have generated it.</p> <p><u>designating</u>: Designating something for a particular Use means specifying it for and restricting it to that Use.</p> <p><u>use</u>: see item #42 above</p> <p><u>device class</u>: The generic name for a group of device types. For example, all display stations belong to the same device class. A device class is different from a device type. A device type is composed of all devices that share a common model number or family (e.g. IBM 4331 printers).</p>  |
| 67. | <p><i>digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class;</i></p> | <p><u>digital signature</u>: see item #66 above</p> <p><u>designating</u>: see item #66 above</p> <p><u>use</u>: see item #42 above</p> <p><u>device class</u>: see item #66 above</p> <p><u>tamper resistance</u>: Making tampering more difficult and/or allowing detection of tampering.</p> <p>In this definition, "tampering" means using (e.g., observing or altering) in any unauthorized manner, or interfering with authorized use.</p> <p><u>digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class</u>: Normal English, incorporating the separately defined terms: generating a Digital Signature</p> | <p><u>digital signature</u>: see item #66 above</p> <p><u>designating</u>: see item #66 above</p> <p><u>use</u>: see item #42 above</p> <p><u>device class</u>: see item #66 above</p> <p><u>tamper resistance</u>: The ability of a <b>Tamper Resistant Barrier</b> to prevent Access, observation, and interference with information or processing encapsulated by the barrier.</p> <p>For the purposes of the construction of "Tamper Resistance," "Tamper/Tampering" is defined as: Using (e.g., observing or altering) in any unauthorized manner, or interfering with authorized use.</p> <p>For the purposes of the construction of "Tamper Resistance," "Access" is defined as set forth in item #4, above.</p> <p><u>digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device</u></p> |

| <u>'721 Claim 1</u> | <u>IT Construction</u>   | <u>MS Construction</u>  |
|---------------------|--|---|
|                     | <p>for the second load module, the Digital Signature Designating that the second load module is for use by a second Device Class. This element further requires that the second Device Class have a different Tamper Resistance or security level than the first Device Class.</p> | <p><u>class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class:</u> (1) <b>Digitally Signing</b> a different ("second") <i>Load Module</i> by using a different ("second") <b>Digital Signature</b> as the signature <i>Key</i>, which signing indicates to any and all devices in the second <b>Device Class</b> that the signor authorized and restricted this <i>Load Module</i> for Use by that device. (2) No VDE device can perform any execution of any <i>Load Module</i> without such authorization. The method ensures that the <i>Load Module</i> cannot execute in a particular <b>Device Class</b> and ensures that no device in that <b>Device Class</b> has the <i>Key(s)</i> necessary to verify the <b>Digital Signature</b>. (3) All devices in the first <b>Device Class</b> have the same persistent (not just occasional) and identified level of <b>Tamper Resistance</b> and the same persistent and identified level of security. All devices in the second <b>Device Class</b> have the same persistent and identified level of <b>Tamper Resistance</b> and same persistent and identified level of security. (4) The identified level of <b>Tamper Resistance</b> or identified level of security (or both) for the first <b>Device Class</b>, is greater than or less than the identified level of <b>Tamper Resistance</b> or identified level of security for the second <b>Device Class</b>.</p> <p>For the purposes of the construction of this phrase, a "<i>Load Module</i>" is defined as set forth in item #4 and "<i>Key</i>" is defined as set forth in item #66, above.</p> |

|     | <u>'721 Claim 1</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|---|---|---|
| 68. | distributing the first load module for <b>use</b> by at least one device in the first <b>device class</b> ; and | <u>use</u> : see item #42 above<br><br><u>device class</u> : see item #66 above | <u>use</u> : see item #42 above<br><br><u>device class</u> : see item #66 above |
| 69. | distributing the second load module for <b>use</b> by at least one device in the second <b>device class</b> .   | <u>use</u> : see item #42 above<br><br><u>device class</u> : see item #66 above | <u>use</u> : see item #42 above<br><br><u>device class</u> : see item #66 above |

|     | <u>'721 Claim 34</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|--|---|---|
| 70. | 34. A <b>protected processing environment</b> comprising:              | <p>The claim contains no requirement of a VDE</p> <p><u>protected processing environment</u>: see item #62 above</p> <p>"Protected processing environment" appears in the preamble of this claim. InterTrust reserves the right to assert that it should not be defined, other than as requiring the individual claim elements.</p> | <p><u>Claim as a Whole</u>: The "<b>Protected Processing Environment</b>" is part of and within VDE. (See item #86 for Microsoft's construction of VDE.)</p> <p><u>protected processing environment</u>: see item #62 above</p>   |
| 71. | a first <b>tamper resistant barrier</b> having a first security level, | <p><u>tamper resistant barrier</u>: Hardware and/or software that provides Tamper Resistance.</p>   | <p><u>tamper resistant barrier</u>: (1) An active device that encapsulates and separates a <b>Protected Processing Environment</b> from the rest of the world.</p> <p>(2) It prevents information and processes within the <b>Protected Processing Environment</b> from being observed, interfered with, and leaving except under appropriate conditions ensuring security.</p> <p>(3) It also <b>Controls</b> external access to the encapsulated <b>Secure</b> resources, processes and information.</p> <p>(4) A <b>Tamper Resistant Barrier</b> is capable of destroying protected information in response to <i>Tampering</i> attempts.</p> <p>For the purposes of the construction of "Tamper Resistant Barrier," "<i>Tamper/Tampering</i>" is defined as set forth in item #67, above.</p> |
| 72. | a first <b>secure</b> execution space, and                             | <p><u>secure</u>: see item #3 above</p>   | <p><u>secure</u>: see item #3 above</p>   |



|     | <u>'721 Claim 34</u>   | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|--|--|---|
| 73. | at least one arrangement within the first <b>tamper resistant barrier</b> that prevents the first <b>secure</b> execution space from executing the same <b>executable</b> accessed by a second <b>secure</b> execution space having a second <b>tamper resistant barrier</b> with a second security level different from the first security level. | <p><u>tamper resistant barrier</u>: see item #71 above</p> <p><u>secure</u>: see item #3 above</p> <p><u>executable</u>: A computer program that can be run, directly or through interpretation.</p> | <p><u>tamper resistant barrier</u>: see item #71 above</p> <p><u>secure</u>: see item #3 above</p> <p><u>executable</u>: A cohesive series of machine code instructions in a format that can be loaded into memory and run (executed) by a connected processor.</p> |

|     | <u>'861 Claim 58</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|--|---|---|
| 74. | 58. A method of creating a first <b>secure container</b> , said method including the following steps;  | The claim contains no requirement of a VDE.<br><br><u>secure container</u> : see item #57 above | <u>Claim as a whole</u> : The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.)<br><br><u>secure container</u> : see item #57 above |
| 75. | accessing a descriptive data structure, said descriptive data structure including or addressing  |   |   |
| 76. | organization information at least in part describing a required or desired organization of a content section of said first <b>secure container</b> , and       | <u>secure container</u> : see item #57 above  | <u>secure container</u> : see item #57 above  |
| 77. | metadata information at least in part specifying at least one step required or desired in creation of said first <b>secure container</b> ;                     | <u>secure container</u> : see item #57 above  | <u>secure container</u> : see item #57 above  |
| 78. | using said descriptive data structure to organize said first <b>secure container</b> contents;   | <u>secure container</u> : see item #57 above  | <u>secure container</u> : see item #57 above  |
| 79. | using said metadata information to at least in part determine specific information required to be included in said first <b>secure container</b> contents; and | <u>secure container</u> : see item #57 above  | <u>secure container</u> : see item #57 above  |

|     | <u>'861 Claim 58</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|--|---|---|
| 80. | generating or identifying at least one rule designed to <b>control</b> at least one <b>aspect</b> of access to or <b>use</b> of at least a portion of said first <b>secure container</b> contents. | <u>control (controlling)</u> : see item #7 above<br><u>aspect</u> : see item #60 above<br><u>use</u> : see item #42 above<br><u>secure container</u> : see item #57 above | <u>control (controlling)</u> : see item #7 above<br><u>aspect</u> : see item #60 above<br><u>use</u> : see item #42 above<br><u>secure container</u> : see item #57 above |

|     | <u>'891 Claim 1</u>  | <u>IT Construction</u>   | <u>MS Construction</u>  |
|-----|--|--|---|
| 81. | 1. A method for using at least one resource processed in a <b>secure</b> operating environment at a first appliance, said method comprising:   | The claim contains no requirement of a VDE.<br><br><u>secure</u> : see item #3 above   | <u>Claim as a whole</u> : The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.)<br><br><u>secure</u> : see item #3 above  |
| 82. | <b>securely</b> receiving a first entity's <b>control</b> at said first appliance, said first entity being located remotely from said operating environment and said first appliance;  | <u>securely (secure)</u> : see item #3 above<br><br><u>control</u> : see item #4 above   | <u>securely (secure)</u> : see item #3 above<br><br><u>control</u> : see item #4 above  |
| 83. | <b>securely</b> receiving a second entity's <b>control</b> at said first appliance, said second entity being located remotely from said operating environment and said first appliance, said second entity being different from said first entity; and | <u>securely (secure)</u> : see item #3 above<br><br><u>control</u> : see item #4 above   | <u>securely (secure)</u> : see item #3 above<br><br><u>control</u> : see item #4 above  |
| 84. | <b>securely</b> processing a data item at said first appliance, using at least one resource, including   | <u>securely (secure)</u> : see item #3 above   | <u>securely (secure)</u> : see item #3 above  |
| 85. | <b>securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item.</b>  | <u>securely (secure)</u> : see item #3 above<br><br><u>use</u> : see item #42 above<br><br><u>control</u> : see item #4 above<br><br><u>securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item</u> : Normal English, incorporating the separately defined terms: the first entity's Control | <u>securely (secure)</u> : see item #3 above<br><br><u>use</u> : see item #42 above<br><br><u>control</u> : see item #4 above<br><br><u>securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item</u> : (1) Processing the resource (component part of a first appliance's Secure |

|  | <u>'891 Claim 1</u> | <u>IT Construction</u>  | <u>MS Construction</u>  |
|--|---------------------|---|---|
|  |                     | <p>and the second entity's Control are Securely applied to govern Use of the data item, the act of Securely applying involving use of the resource.</p> | <p>Operating Environment) within the Secure Operating Environment's special-purpose Secure Processing Unit (SPU) to execute the first <b>Control</b> and second <b>Control</b> in combination within the SPU.</p> <p>(2) This execution of these <b>Controls</b> governs (<b>Controls</b>) all Use of the data item by all users, processes, and devices.</p> <p>(3) The processing of the resource and execution of the <b>Controls</b> cannot be observed from outside the SPU and is performed only after the integrity of the resource and <b>Controls</b> is cryptographically verified.</p> <p>(4) A Secure Processing Unit is a special-purpose unit isolated from the rest of the world in which a hardware <b>Tamper Resistant Barrier</b> encapsulates a processor and internal <b>Secure</b> memory.</p> <p>(5) The processor cryptographically verifies the integrity of all code loaded from the <b>Secure</b> memory prior to execution, executes only the code that the processor has authenticated for its Use, and is otherwise <b>Secure</b>.</p> |

|     | <u>'900 Claim 155</u>                              | <u>IT Construction</u>   | <u>MS Construction</u>   |
|-----|--|--|--|
| 86. | 155. A virtual distribution environment comprising | <p><u>Virtual Distribution Environment</u>: This term is contained in the preamble of the claim and should not be defined, other than as requiring the individual claim elements.</p> <p>Without waiving its position that no separate definition is required, if required to propose such a definition, InterTrust proposes the following: secure, distributed electronic transaction management and rights protection system for controlling the distribution and/or other usage of electronically provided and/or stored information.</p> | <p><u>Claim as a Whole</u>: The "virtual distribution environment" is VDE.</p> <p><u>Virtual Distribution Environment</u>: (1) <u>Data Security and Commerce World</u>: InterTrust's February 13, 1995, patent application described as its "invention" a <b>Virtual Distribution Environment</b> ("VDE invention") for securing, administering, and auditing all security and commerce digital information within its multi-node world (community). VDE guarantees to all VDE "participants" identified in the patent application that it will limit all <i>Access</i> to and <i>Use</i> (i.e., interaction) of such information to authorized activities and amounts, will ensure any requested reporting of and payment for such <i>Use</i>, and will maintain the availability, secrecy, integrity, non-repudiation and authenticity of all such information present at any of its nodes (including protected content, information about content usage, and content <b>Controls</b>).</p> <p>VDE is <b>Secure</b> against at least the threats identified in the February 1995, patent application to this availability (no user may delete the information without authorization), secrecy (neither available nor disclosed to unauthorized persons or processes), integrity (neither intentional nor accidental alteration), non-repudiation (neither the receiver can disavow the receipt of a message nor can the sender disavow the origination of that message) and authenticity (asserted characteristics are genuine). VDE further provides and requires the components and capabilities described below. Anything less than or different than this is not VDE or the described "invention."</p> |

| <u>'900 Claim 155</u> | <u>IT Construction</u> | <u>MS Construction</u>  |
|-----------------------|------------------------|---|
|                       |                        | <p>(2) <u>Secure Processing Environment</u>: At each node where <b>VDE</b>-protected information is <i>Accessed</i>, <i>Used</i>, or assigned control information, <b>VDE</b> requires a <i>Secure Processing Environment</i> (as set forth in item #6).</p> <p>(3) <u>VDE Controls</u>: <b>VDE Allows Access</b> to or <i>Use</i> of protected information and processes only through execution of (and satisfaction of the requirements imposed by) <b>VDE Control(s)</b>.</p> <p>(4) <u>VDE Secure Container</u>: See construction of <b>Secure Container</b> (see item #57).</p> <p>(5) <u>Non-Circumventable</u>: <b>VDE</b> is non-circumventable (sequestered). It intercepts all attempts by any and all users, processes, and devices, to <i>Access</i> or <i>Use</i>, such as observing, interfering with, or removing) protected information, and prevents all such attempts other than as allowed by execution of (and satisfaction of all requirements imposed by) associated <b>VDE Controls</b> within <i>Secure Processing Environment(s)</i>.</p> <p>(6) <u>Peer to Peer</u>: <b>VDE</b> is peer-to-peer. Each <b>VDE</b> node has the innate ability to perform any role identified in the patent application (e.g., end user, content packager, distributor, <b>Clearinghouse</b>, etc.), and can protect information flowing in any direction between any nodes. <b>VDE</b> is not client-server. It does not pre-designate and restrict one or more nodes to act solely as a "server" (a provider of information (e.g., authored content, control information, etc.) to other nodes) or "client" (a requestor of such information). All types of protected-content transactions can proceed without requiring interaction with any server.</p> |

|  | <u>'900 Claim 155</u> | <u>IT Construction</u> | <u>MS Construction</u>   |
|--|-----------------------|------------------------|--|
|  |                       |                        | <p>(7) <u>Comprehensive Range of Functions</u>: VDE comprehensively governs (<b>Controls</b>) all security and commerce activities identified in the patent application, including (a) metering, budgeting, monitoring, reporting, and auditing information usage, (b) billing and paying for information usage, and (c) negotiating, signing and enforcing contracts that establish users' rights to <i>Access</i> or <i>Use</i> information.</p> <p>(8) <u>User-Configurable</u>: The specific protections governing (<b>Controlling</b>) specific VDE-protected information are specified, modified, and negotiated by VDE's users. For example, VDE enables a consumer to place limits on the nature of content that may be <i>Accessed</i> at her node (e.g., no R-rated material) or the amount of money she can spend on viewing certain content, both subject only to other users' senior <b>Controls</b>.</p> <p>(9) <u>General Purpose; Universal</u>: VDE is universal as opposed to being limited to or requiring any particular type of appliance, information, or commerce model. It is a single, unified standard and environment within which an unlimited range of electronic rights protection, data security, electronic currency, and banking applications can run.</p> <p>(10) <u>Flexible</u>: VDE is more flexible than traditional information security and commerce systems. For example, VDE allows consumers to pay for only the user-defined portion of information that the user actually uses, and to pay only in proportion to any quantifiable VDE event (e.g., for only the number of paragraphs displayed from a book), and allows editing the content in VDE containers while maintaining its security.</p> |



|     | <u>'900 Claim 155</u>  | <u>IT Construction</u>   | <u>MS Construction</u>   |
|-----|--|--|--|
|     |  |  | For the purposes of the construction of "VDE," " <i>Secure Processing Environment</i> " and "Access" are defined as set forth in item #4, above.   |
| 87. | a first <b>host processing environment</b> comprising                                    | <p><u>host processing environment</u>: This term is explicitly defined in the claim and therefore needs no additional definition. It consists of those elements listed in the claim.</p> <p>Without waiving its position that no separate definition is required, if required to propose such a definition, InterTrust proposes the following: a Protected Processing Environment incorporating software-based security.</p> | <p><u>host processing environment</u>: (1) A processing environment within a VDE node which is not a <i>Secure Processing Environment</i>.</p> <p>(2) A "host processing environment" may either be "secure" or "not secure."</p> <p>(3) A "secure host processing environment" is a self-contained <b>Protected Processing Environment</b>, formed by loaded, <b>Executable</b> programming executing on a general purpose CPU (not a Secure Processing Unit ) running in protected (privileged) mode.</p> <p>(4) A "non-secure host processing environment" is formed by loaded, <b>Executable</b> programming executing on a general purpose CPU (not a Secure Processing Unit) running in user mode.</p> <p>For the purposes of the construction of "Host Processing Environment," a "<i>Secure Processing Environment</i>" is defined as set forth in item #4, above.</p> |
| 88. | a central processing unit;   |  |  |
| 89. | main memory operatively connected to said central processing unit;                       |  |  |
| 90. | mass storage operatively connected to said central processing unit and said main memory; |  |  |

|     | <u>'900 Claim 155</u>  | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|--|---|---|
| 91. | said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising: |   |   |
| 92. | machine check programming which <i>derives information from one or more aspects of said host processing environment</i> ,  | <p><u>derives</u>: Normal English: obtains, receives or arrives at through a process of reasoning or deduction. In the context of computer operations, the "process of reasoning or deduction" constitutes operations carried out by the computer.</p> <p><u>aspect</u>: see item #60 above</p> <p><u>host processing environment</u>: see item #87 above</p> <p><u>derives information from one or more aspects of said host processing environment</u>: Normal English, incorporating the separately defined terms: Derives (including creates) information based on at least one Aspect of the previously referred to Host Processing Environment.</p> | <p><u>derives</u>: To retrieve from a specified source.</p> <p><u>aspect</u>: see item #60 above</p> <p><u>host processing environment</u>: see item #87 above</p> <p><u>derives information from one or more aspects of said host processing environment</u>: (1) <b>Deriving from the Host Processing Environment</b> hardware one or more values that uniquely and persistently identify the <b>Host Processing Environment</b> and distinguish it from other <b>Host Processing Environments</b>. (2) The "one or more aspects of said host processing environment" are persistent elements or properties of the <b>Host Processing Environment</b> itself that are capable of being used to distinguish it from other environments, as opposed to, e.g., data or programs stored within the mass storage or main memory, or processes executing within the <b>Host Processing Environment</b>.</p> |
| 93. | one or more storage locations storing said information;  |   |   |

|     | <u>'900 Claim 155</u>   | <u>IT Construction</u>  | <u>MS Construction</u>  |
|-----|---|---|---|
| 94. | integrity programming which causes said machine check programming to <b>derive</b> said information, <b>compares</b> said information to information previously stored in said one or more storage locations, and | <u>derive</u> : see item #92 above<br><br><u>compares</u> : Normal English: examines for the purpose of noting similarities and differences. "Comparison" refers to the act of comparing. | <u>derive</u> : see item #92 above<br><br><u>compares</u> : A processor operation that evaluates two quantities and sets one of three flag conditions as a result of the comparison – greater than, less than, or equal to. |
| 95. | generates an indication based on the result of said <b>comparison</b> ; and   | <u>comparison (compares)</u> : see item #94 above   | <u>comparison (compares)</u> : see item #94 above   |
| 96. | programming which takes one or more actions based on the state of said indication;  |   |   |
| 97. | said one or more actions including at least temporarily halting further processing.   |   |   |

|      | <u>'912 Claim 8</u>   | <u>IT Construction</u>  | <u>MS Construction</u>   |
|------|---|---|--|
| 98.  | 8. A process comprising the following steps:  | The claim contains no requirement of a VDE.   | <u>Claim as a whole:</u> The recited method is performed within a VDE. (See item #93 for Microsoft's construction of VDE.)   |
| 99.  | accessing a first record <b>containing</b> information directly or indirectly identifying one or more elements of a first <b>component assembly</b> , | <u>containing:</u> see item #58 above<br><br><u>component assembly:</u> Components are code and/or data elements that are independently deliverable. A Component Assembly is two or more components associated together. Component Assemblies are utilized to perform operating system and/or applications tasks. | <u>containing:</u> see item #58 above<br><br><u>component assembly:</u> (1) A cohesive <b>Executable</b> component created by a channel which binds or links together two or more independently deliverable <i>Load Modules</i> , and associated data.<br>(2) A <b>Component Assembly</b> is assembled, and executes, only within a <b>VDE Secure Processing Environment</b> .<br>(3) A <b>Component Assembly</b> is assembled dynamically in response to, and to service, a particular content-related activity (e.g., a particular Use request).<br>(4) Each <b>VDE Component Assembly</b> is assigned and dedicated to a particular activity, particular user(s), and particular protected information.<br>(5) Each <b>Component Assembly</b> is independently assembled, loadable and deliverable vis-à-vis other <b>Component Assemblies</b> .<br>(6) The dynamic assembly of a <b>Component Assembly</b> is directed by a "blueprint" <b>Record Containing</b> control information for this particular activity on this particular information by this particular user(s).<br>(7) <b>Component Assemblies</b> are extensible and can be configured and reconfigured (modified) by all users, and combined by all users with other <b>Component Assemblies</b> , subject only to other users' "senior" <b>Controls</b> .<br><br>For the purposes of the construction of "Component Assembly," " <i>Load Module</i> ," " <i>Secure Processing Environment</i> " and " <i>Record</i> " are defined as set forth in item #4 above. |
| 100. | at least one of said elements including at least some   | <u>executable programming (executable):</u> see item #73 above  | <u>executable programming:</u> A cohesive series of machine code instructions, comprising a computer program, in a   |

|      | <u>'912 Claim 8</u>   | <u>IT Construction</u>   | <u>MS Construction</u>  |
|------|---|--|---|
|      | executable programming,   |  | format that can be loaded into memory and run (executed) by a connected processor. A "computer program" is a complete series of definitions and instructions that when executed on a computer will perform a required or requested task.  |
| 101. | at least one of said elements constituting a load module,   |  |   |
| 102. | said load module including executable programming and a header;   | <u>executable programming (executable):</u><br>see item #73 above  | <u>executable programming:</u> see item #100 above  |
| 103. | said header including an execution space identifier identifying at least one aspect of an execution space required for use and/or execution of the load module associated with said header; | <u>identifier:</u> see item #28<br><br><u>aspect:</u> see item #59 above<br><br><u>use:</u> see item #42 above<br><br><u>identifying at least one aspect of an execution space required for use and/or execution of the load module:</u><br>Normal English, incorporating the separately defined terms: identifying an Aspect (e.g. security level) of an execution space that is needed in order for the load module to execute or otherwise be used. | <u>identifier:</u> see item #28<br><br><u>aspect:</u> see item #59 above<br><br><u>use:</u> see item #42 above<br><br><u>identifying at least one aspect of an execution space required for use and/or execution of the load module:</u><br>(1) Defining fully, without reference to any other information, at least one of the persistent elements or properties (Aspects) (that are capable of being used to distinguish it from other environments of an execution space) that are required for any Use, and/or for any execution, of the <i>Load Module</i> .<br>(2) An execution space without all of those required aspects is incapable of making any such execution and/or other Use (e.g., Copying, displaying, printing) of the <i>Load Module</i> .<br><br>For the purposes of the construction of this phrase, a " <i>Load Module</i> " is defined as set forth in item #4, above |

|      | <b><u>'912 Claim 8</u></b>  | <b><u>IT Construction</u></b>   | <b><u>MS Construction</u></b>   |
|------|---|---|---|
| 104. | said execution space <b>identifier</b> provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security; | <u>identifier</u> : see item #28  | <u>identifier</u> : see item #28  |
| 105. | using said information to identify and locate said one or more elements;  |   |   |
| 106. | accessing said located one or more elements;  |   |   |
| 107. | <b>securely</b> assembling said one or more elements to form at least a portion of said first <b>component assembly</b> ;   | <u>securely</u> : see item #3 above<br><br><u>component assembly</u> : see item #98 above | <u>securely</u> : see item #3 above<br><br><u>component assembly</u> : see item #98 above |
| 108. | executing at least some of said <b>executable programming</b> ; and   | <u>executable programming (executable)</u> : see item #73 above                           | <u>executable programming</u> : see item #100 above                                       |
| 109. | checking said record for validity prior to performing said executing step.  |   |   |

|      | <u>'912 Claim 35</u>  | <u>IT Construction</u>   | <u>MS Construction</u>   |
|------|---|--|--|
| 110. | 35. A process comprising the following steps:   | The claim contains no requirement of a VDE.  | <u>Claim as a whole:</u> The recited method is performed within a VDE. (See item #86 for Microsoft's construction of VDE.) |
| 111. | at a first processing environment receiving a first record from a second processing environment remote from said first processing environment;                |  |  |
| 112. | said first record being received in a <b>secure container</b> ;   | <u>secure container</u> : see item #57 above   | <u>secure container</u> : see item #57 above   |
| 113. | said first record <b>containing</b> identification information directly or indirectly identifying one or more elements of a first <b>component assembly</b> ; | <u>containing</u> : see item #57 above<br><br><u>component assembly</u> : see item #98 above | <u>containing</u> : see item #57 above<br><br><u>component assembly</u> : see item #98 above                               |
| 114. | at least one of said elements including at least some <b>executable programming</b> ;   | <u>executable programming (executable)</u> : see item #73 above                              | <u>executable programming</u> : see item #100 above  |
| 115. | said <b>component assembly</b> allowing access to or <b>use</b> of specified information;   | <u>component assembly</u> : see item #98 above<br><br><u>use</u> : see item #42 above        | <u>component assembly</u> : see item #98 above<br><br><u>use</u> : see item #42 above                                      |
| 116. | said <b>secure container</b> also including a first of said elements;   | <u>secure container</u> : see item #57 above   | <u>secure container</u> : see item #57 above   |
| 117. | accessing said first record;  |  |  |
| 118. | using said identification information to identify and locate  |  |  |

|      | <u>'912 Claim 35</u>  | <u>IT Construction</u>   | <u>MS Construction</u>   |
|------|---|--|--|
|      | said one or more elements;  |  |  |
| 119. | said locating step including locating a second of said elements at a third processing environment located remotely from said first processing environment and said second processing environment; |  |  |
| 120. | accessing said located one or more elements;  |  |  |
| 121. | said element accessing step including retrieving said second element from said third processing environment;  |  |  |
| 122. | <b>securely</b> assembling said one or more elements to form at least a portion of said first <b>component assembly</b> specified by said first record; and                                       | <u>securely (secure)</u> : see item #3 above<br><u>component assembly</u> : see item #98 above | <u>securely (secure)</u> : see item #3 above<br><u>component assembly</u> : see item #98 above |
| 123. | executing at least some of said <b>executable programming</b> .   | <u>executable programming (executable)</u> : see item #73 above                                | <u>executable programming</u> : see item #100 above  |
| 124. | said executing step taking place at said first processing environment.  |  |  |



## EXHIBIT B

### PLR 4-3(b) – The Parties’ Construction of Disputed Terms & Phrases

|    | Claim<br>Term/Phrase                              | InterTrust Construction  | Microsoft Construction   |
|----|---|--|--|
| 1. | aspect<br><br>683.2<br>861.58<br>900.155<br>912.8 | Feature, element, property or state.   | An aspect of an environment is a persistent element or property of that environment that can be used to distinguish it from other environments.  |
| 2. | authentication<br><br>193.15                      | Identifying (e.g., a person, device, organization, document, file, etc.). Includes uniquely identifying or identifying as a member of a group.   | To establish that the following asserted characteristics of something (e.g., a person, device, organization, document, file, etc.) are genuine: its identity, its data integrity, (i.e., it has not been altered) and its origin integrity (i.e., its source and time of origination).   |
| 3. | budget<br><br>193.1                               | Information specifying a limitation on usage.  | (1) A unique type of “method” that specifies a decrementable numerical limitation on future Use (e.g., copying) of digital information and how such Use will be paid for, if at all.<br>(2) A “method” is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements, and/or relationships for use in performing, and/or preparing to perform, basic instructions in relation to the operation of one or more electronic appliances. |
| 4. | clearinghouse<br><br>193.19                       | A provider of financial and/or administrative services for a number of entities; or an entity responsible for the collection, maintenance, and/or distribution of materials, information, licenses, etc. | (1) A computer system that provides intermediate storing and forwarding services for both content and audit information, and which two or more parties trust to provide its services independently because it is operated under constraint of VDE security.<br>(2) “Audit information” means all information created, stored, or reported in connection with an “auditing” process. “Auditing”   |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|    | Claim<br>Term/Phrase                       | InterTrust Construction   | Microsoft Construction   |
|----|--|---|--|
|    |  |   | means tracking, metering and reporting the usage of particular information or a particular appliance.  |
| 5. | compares<br><br>900.155                    | Normal English: examines for the purpose of noting similarities and differences.  | A processor operation that evaluates two quantities and sets one of three flag conditions as a result of the comparison – greater than, less than, or equal to.  |
| 6. | component<br>assembly<br><br>912.8, 912.35 | Components are code and/or data elements that are independently deliverable. A Component Assembly is two or more components associated together. Component Assemblies are utilized to perform operating system and/or applications tasks. | <p>(1) A cohesive <b>Executable</b> component created by a channel which binds or links together two or more independently deliverable <i>Load Modules</i> (see below), and associated data.</p> <p>(2) A <b>Component Assembly</b> is assembled, and executes, only within a <i>VDE Secure Processing Environment</i> (see below).</p> <p>(3) A <b>Component Assembly</b> is assembled dynamically in response to, and to service, a particular content-related activity (e.g., a particular <b>Use</b> request).</p> <p>(4) Each <b>VDE Component Assembly</b> is assigned and dedicated to a particular activity, particular user(s), and particular protected information.</p> <p>(5) Each <b>Component Assembly</b> is independently assembled, loadable and deliverable vis-à-vis other <b>Component Assemblies</b>.</p> <p>(6) The dynamic assembly of a <b>Component Assembly</b> is directed by a “blueprint” <i>Record</i> (see below) <b>Containing</b> control information for this particular activity on this particular information by this particular user(s).</p> <p>(7) <b>Component Assemblies</b> are extensible and can be configured and reconfigured (modified) by all users, and combined by all users with other <b>Component Assemblies</b>,</p> |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction  |
|--|----------------------|-------------------------|---|
|  |                      |                         | <p>subject only to other users' "senior" Controls.</p> <p>For the purposes of the construction of "Component Assembly," a "<i>Load Module</i>" is defined as follows: An <b>Executable</b>, modular unit of machine code (which may include data) suitable for loading into memory for execution by a processor. A load module is encrypted (when not within a secure processing unit) and has an <b>Identifier</b> that a calling process must provide to be able to use the load module. A load module is combinable with other load modules, and associated data, to form <b>Executable Component Assemblies</b>. A load module can execute only in a <b>VDE Protected Processing Environment</b>. Library routines are not load modules and dynamic link libraries are not load modules.</p> <p>For the purposes of the construction of "Component Assembly," a "<i>Secure Processing Environment</i>" is defined as follows: A Secure Processing Environment is uniquely identifiable, self-contained, non-circumventable, and trusted by all other VDE nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the patent application as being protected, and to guarantee that such information will be accessed and <b>Used</b> only as expressly authorized by the associated <b>VDE Controls</b>, and to guarantee that all requested reporting of and payments for protected information use will be made. A Secure Processing</p> |

|    | Claim<br>Term/Phrase   | InterTrust Construction  | Microsoft Construction   |
|----|--|--|--|
|    |  |  | <p>Environment is formed by, and requires, a Secure Processing Unit having a hardware <b>Tamper Resistant Barrier</b> encapsulating a processor and internal <b>Secure</b> memory. The <b>Tamper Resistant Barrier</b> prevents all unauthorized interference, removal, observation, and other Use of the information and processes within it.</p> <p>For the purposes of the construction of "Component Assembly," a "<i>Record</i>" is defined as follows: A data structure that is a collection of fields (elements), each with its own name and type. Unlike an array, whose elements are accessed using an index, the elements of a record are accessed by name. A record can be accessed as a collective unit of elements, or the elements can be accessed individually.</p> |
| 7. | contain<br><br>683.2<br>912.8, 912.35                                  | Normal English: to have within or to hold. In the context of an element contained within a data structure (e.g., a secure container), the contained element may be either directly within the container or the container may hold a reference indicating where the element may be found. | Physically (directly) storing within, as opposed to addressing (i.e., referring to something by the explicitly identified location where it is stored, without directly storing it).   |
| 8. | control (n.)<br><br>193.1, 193.11,<br>193.15, 193.19<br>683.2<br>891.1 | Information and/or programming controlling operations on or use of resources (e.g., content) including (a) permitted, required or prevented operations, (b) the nature or extent of such operations or (c) the consequences of such operations.  | <p>(1) Independent, special-purpose, <b>Executable</b>, which can execute only within a <i>Secure Processing Environment</i>.</p> <p>(2) Each VDE Control is a <b>Component Assembly</b> dedicated to a particular activity (e.g., editing, modifying another <b>Control</b>, a user-defined action, etc.), particular user(s), and particular protected information, and whose satisfactory execution is necessary to <i>Allowing</i> (see below) that activity.</p>  |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction   |
|--|----------------------|-------------------------|--|
|  |                      |                         | <p>(3) Each separate information <i>Access</i> (see below) or <i>Use</i> is independently <b>Controlled</b> by independent <b>VDE Control(s)</b>.</p> <p>(4) Each <b>VDE Control</b> is assembled within a <i>Secure Processing Environment</i> from independently deliverable modular components (e.g., <i>Load Modules</i> or other <b>Controls</b>), dynamically in response to an information <i>Access</i> or <i>Use</i> Request.</p> <p>(5) The dynamic assembly of a <b>Control</b> is directed by a "blueprint" <i>Record</i> (put in place by one or more <b>VDE</b> users) <b>Containing</b> control information identifying the exact modular code components to be assembled and executed to govern (i.e., <b>Control</b>) this particular activity on this particular information by this particular user(s).</p> <p>(6) Each <b>Control</b> is independently assembled, loaded and delivered vis-à-vis other <b>Controls</b>.</p> <p>(7) Control information and <b>Controls</b> are extensible and can be configured and modified by all users, and combined by all users with any other <b>VDE</b> control information or <b>Controls</b> (including that provided by other users), subject only to "senior" user <b>Controls</b>.</p> <p>(8) Users can assign control information (including alternative control information) and <b>Controls</b> to an arbitrarily fine, user-defined portion of the protected information, such as a single paragraph of a document, as opposed to being limited to file-based controls.</p> <p>(9) <b>VDE Controls</b> reliably limit <i>Use</i> of the protected information to only authorized activities and amounts.</p> |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|    | Claim<br>Term/Phrase      | InterTrust Construction   | Microsoft Construction   |
|----|---------------------------|---|--|
|    |                           |   | <p>For the purposes of the construction of "Control," a "<i>Secure Processing Environment</i>" is defined as set forth in item #6, above.</p> <p>For the purposes of the construction of "Control," "<i>Allowing</i>" is defined as follows: Actively permitting an action that otherwise cannot be taken (i.e., is prohibited) by any user, process, or device. In VDE, an action is allowed only through execution (within a <i>Secure Processing Environment</i>) of the <b>VDE Control(s)</b> assigned to the particular action request, and satisfaction of all requirements imposed by such execution.</p> <p>For the purposes of the construction of "Control," "<i>Access</i>" is defined as follows: To satisfactorily perform the steps necessary to obtain something so that it can be <b>Used</b> in some manner (e.g., for information: copied, printed, decrypted, encrypted, saved, modified, observed, or moved, etc.). In VDE, access to protected information is achieved only through execution (within a <i>Secure Processing Environment</i>) of the <b>VDE Control(s)</b> assigned to the particular "access" request, satisfaction of all requirements imposed by such execution, and the <b>Controlled</b> opening of the <b>Secure Container Containing</b> the information.</p> <p>For the purposes of the construction of "Control," "<i>Load Module</i>" and "<i>Record</i>" are defined as set forth in item #6, above.</p> |
| 9. | controlling, control (v.) | Normal English: to exercise authoritative or dominating influence over; direct. | (1) Reliably defining and enforcing the conditions and requirements under which an action that otherwise   |

|     | Claim<br>Term/Phrase  | InterTrust Construction  | Microsoft Construction   |
|-----|---|--|--|
|     | 193.1<br>861.58   |  | <p>cannot be taken, will be <i>Allowed</i>, and the manner in which it may occur. Absent verified satisfaction of those conditions and requirements, the action cannot be taken by any user, process or device.</p> <p>(2) In VDE, an action is <b>Controlled</b> through execution of the applicable <b>VDE Control(s)</b> within a <b>VDE Secure Processing Environment</b>.</p> <p>(3) More specifically, in VDE, <b>Controlling</b> is effected by use of <b>VDE Controls</b>, <b>VDE Secure Containers</b>, and <b>VDE foundation</b> (including <b>VDE Secure Processing Environment</b>, "object registration," and other mechanisms for allegedly individually ensuring that specific <b>Controls</b> are enforced vis-à-vis specific objects (and their content at an arbitrary granular level) and specific "users").</p> <p>For the purposes of the construction of "Control (v.)" et al, "<i>Allowed</i>" is defined as set forth in item #8, above, and "<i>Secure Processing Environment</i>" is defined as set forth in item #6, above.</p> |
| 10. | copy, copied, copying<br><br>193.1, 193.11,<br>193.15, 193.19 | Reproduce, reproduced, reproducing. The reproduction must be usable, may incorporate all of the original item or only some of it, and may involve some changes to the item as long as the essential nature of the content remains unchanged. | <p>(1) To reproduce all of a <i>Digital File</i> or other complete physical block of data from one location on a storage medium to another location on the same or different storage medium, leaving the original block of data unchanged, such that two distinct and independent objects exist.</p> <p>(2) Although the layout of the data values in physical storage may differ from the original, the resulting "copy" is logically indistinguishable from the original.</p> <p>(3) The resulting "copy" may or may not be encrypted, ephemeral, usable, or accessible.</p>   |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|     | Claim<br>Term/Phrase                                 | InterTrust Construction   | Microsoft Construction  |
|-----|--|---|---|
|     |  |   | For the purposes of the construction of "Copy," et al, a " <i>Digital File</i> " is defined as: A named, static unit of storage allocated by a "file system" and <b>Containing</b> digital information. A digital file enables any application using the "file system" to randomly access its contents and to distinguish it by name from every other such unit. A copy of a digital file is a separate digital file. A "file system" is the portion of the operating system that translates requests made by application programs for operations on "files" into low-level tasks that can control storage devices such as disk drives. |
| 11. | derive<br><br>900.155                                | Normal English: obtain, receive or arrive at through a process of reasoning or deduction. In the context of computer operations, the "process of reasoning or deduction" constitutes operations carried out by the computer.                  | To retrieve from a specified source.  |
| 12. | designating<br><br>721.1                             | Normal English: indicating, specifying, pointing out or characterizing.   | Designating something for a particular Use means specifying it for and restricting it to that Use.  |
| 13. | device class<br>721.1                                | A group of devices which share at least one attribute.  | The generic name for a group of device types. For example, all display stations belong to the same device class. A device class is different from a device type. A device type is composed of all devices that share a common model number or family (e.g. IBM 4331 printers).  |
| 14. | digital signature,<br>digitally signing<br><br>721.1 | digital signature: A digital value, verifiable with a key, that can be used to determine the source and/or integrity of a signed item (e.g., a file, program, etc.).<br><br>Digitally signing is the process of creating a digital signature. | <u>digital signature</u> : A computationally unforgeable string of characters (e.g., bits) generated by a cryptographic operation on a block of data using some secret. The string can be generated only by an entity that knows the secret, and hence provides evidence that the   |



|     | Claim<br>Term/Phrase  | InterTrust Construction   | Microsoft Construction   |
|-----|---|---|--|
|     |   |   | <p>entity must have generated it.</p> <p><u>digitally signing</u>:</p> <p>(1) Creating a <b>Digital Signature</b> using a secret <i>Key</i> (see below).</p> <p>(2) In symmetric key cryptography, a “secret key” is a <i>Key</i> that is known only to the sender and recipient. In asymmetric key cryptography, a “secret key” is the private <i>Key</i> of a public/private key pair, in which the two keys are related uniquely by a predetermined mathematical relationship such that it is computationally infeasible to determine one from the other.</p> <p>For the purposes of the construction of “Digital Signature” and “Digital Signing,” a “<i>Key</i>” is defined as: A bit sequence used and needed by a cryptographic algorithm to encrypt a block of plain text or to decrypt a block of cipher text. A key is different from a key seed or other information from which the actual encryption and/or decryption key is constructed, <b>Derived</b>, or otherwise identified. In symmetric key cryptography, the same key is used for both encryption and decryption. In asymmetric or “public key” cryptography, two related keys are used; a block of text encrypted by one of the two keys (e.g., the “public key”) can be decrypted only by the corresponding key (e.g., the “private key”).</p> |
| 15. | <p>executable programming, executable</p> <p>721.34<br/>912.8, 912.35</p> | A computer program that can be run, directly or through interpretation. | <p><u>executable</u>: A cohesive series of machine code instructions in a format that can be loaded into memory and run (executed) by a connected processor.</p>   |

|     | Claim<br>Term/Phrase                       | InterTrust Construction  | Microsoft Construction   |
|-----|--|--|--|
|     |  |  | <u>executable programming</u> : A cohesive series of machine code instructions, comprising a computer program, in a format that can be loaded into memory and run (executed) by a connected processor. A "computer program" is a complete series of definitions and instructions that when executed on a computer will perform a required or requested task.   |
| 16. | host processing environment<br><br>900.155 | <p>This term is explicitly defined in the claim and therefore needs no additional definition. It consists of those elements listed in the claim.</p> <p>Without waiving its position that no separate definition is required, if required to propose such a definition, InterTrust proposes the following: a Protected Processing Environment incorporating software-based security.</p> | <p>(1) A processing environment within a VDE node which is not a <i>Secure Processing Environment</i>.</p> <p>(2) A "host processing environment" may either be "secure" or "not secure."</p> <p>(3) A "secure host processing environment" is a self-contained <b>Protected Processing Environment</b>, formed by loaded, <b>Executable</b> programming executing on a general purpose CPU (not a Secure Processing Unit ) running in protected (privileged) mode.</p> <p>(4) A "non-secure host processing environment" is formed by loaded, <b>Executable</b> programming executing on a general purpose CPU (not a Secure Processing Unit) running in user mode.</p> <p>For the purposes of the construction of "host processing environment," a "<i>Secure Processing Environment</i>" is defined as set forth in item #6, above.</p> |
| 17. | identifier<br><br>193.15<br>912.8          | <p>Information used to identify something or someone (e.g., a password).</p> <p>In this definition, "identify" means to establish the identity of or to ascertain the origin, nature, or</p>   | <p>Any text string used as a label naming an individual instance of what it <i>Identifies</i>.</p> <p>For the purpose of the construction of "Identifier," "Identify" is defined as: To establish as being a particular</p>  |

|     | Claim<br>Term/Phrase                                    | InterTrust Construction  | Microsoft Construction  |
|-----|---|--|---|
|     |   | definitive characteristics of; includes identifying as an individual or as a member of a group.  | instance of a person or thing.  |
| 18. | protected processing environment<br><br>683.2<br>721.34 | <p>An environment in which processing and/or data is at least in part protected from tampering. The level of protection can vary, depending on the threat.</p> <p>In this definition, "environment" means capabilities available to a program running on a computer or other device or to the user of a computer or other device. Depending on the context, the environment may be in a single device (e.g., a personal computer) or may be spread among multiple devices (e.g., a network).</p> | <p>(1) A uniquely identifiable, self-contained computing base trusted by all <b>VDE</b> nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the February, 1995, patent application as being protected, and to guarantee that such information will be <i>Accessed</i> and <b>Used</b> only as expressly authorized by <b>VDE Controls</b>.</p> <p>(2) At most <b>VDE</b> nodes, the <b>Protected Processing Environment</b> is a <i>Secure Processing Environment</i> which is formed by, and requires, a hardware <b>Tamper Resistant Barrier</b> encapsulating a special-purpose Secure Processing Unit having a processor and internal secure memory. "Encapsulated" means hidden within an object so that it is not directly accessible but rather is accessible only through the object's restrictive interface.</p> <p>(3) The <b>Tamper Resistant Barrier</b> prevents all unauthorized (intentional or accidental) interference, removal, observation, and use of the information and processes within it, by all parties (including all users of the device in which the <b>Protected Processing Environment</b> resides), except as expressly authorized by <b>VDE Controls</b>.</p> <p>(4) A <b>Protected Processing Environment</b> is under <b>Control</b> of <b>Controls</b> and control information provided by one or more parties, rather than being under <b>Control</b> of the appliance's users or programs.</p> <p>(5) Where a <b>VDE</b> node is an</p> |

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction  |
|--|----------------------|-------------------------|---|
|  |                      |                         | <p>established financial Clearinghouse, or other such facility employing physical facility and user-identity Authentication security procedures trusted by all VDE nodes, and the VDE node does not Access or Use VDE-protected information, or assign VDE control information, then the Protected Processing Environment at that VDE node may instead be formed by a general-purpose CPU that executes all VDE "security" processes in protected (privileged) mode.</p> <p>(6) A Protected Processing Environment requires more than just verifying the integrity of Digitally Signed Executable programming prior to execution of the programming; or concealment of the program, associated data, and execution of the program code; or use of a password as its protection mechanism.</p> <p>For the purposes of the construction of "Protected Processing Environment," a "Secure Processing Environment" is defined as set forth in item #6, above, and "Access" is defined as set forth in item #8, above.</p> |

|     | <b>Claim<br/>Term/Phrase</b>  | <b>InterTrust Construction</b>   | <b>Microsoft Construction</b>   |
|-----|---|--|---|
| 19. | secure, securely<br><br>193.1, 193.11,<br>193.15<br>683.2<br>721.34<br>861.58<br>891.1<br>912.8, 912.35 | One or more mechanisms are employed to prevent, detect or discourage misuse of or interference with information or processes. Such mechanisms may include concealment, Tamper Resistance, Authentication and access control. Concealment means that it is difficult to read information (for example, programs may be encrypted). Tamper Resistance and Authentication are separately defined. Access control means that access to information or processes is limited on the basis of authorization. Security is not absolute, but is designed to be sufficient for a particular purpose. | (1) A state in which all users of a system are guaranteed that all information, processes, and devices within the system, shall have their availability, secrecy, integrity, authenticity and nonrepudiation maintained against all of the identified threats thereto.<br>(2) "Availability" means the property that information is accessible and usable upon demand by authorized persons, at least to the extent that no user may delete the information without authorization.<br>(3) "Secrecy," also referred to as confidentiality, means the property that information (including computer processes) is not made available or disclosed to unauthorized persons or processes.<br>(4) "Integrity" means the property that information has not been altered either intentionally or accidentally.<br>(5) "Authenticity" means the property that the characteristics asserted about a person, device, program, information, or process are genuine and timely, particularly as to identity, data integrity, and origin integrity.<br>(6) "Nonrepudiation" means the property that a sender of information cannot deny its origination and that a recipient of information cannot deny its receipt. |
| 20. | secure container<br><br>683.2<br>861.58<br>912.35   | A container that is Secure.<br><br>In this definition, "container" means a digital file containing linked and/or embedded items.   | (1) A VDE Secure Container is a self-contained, self-protecting data structure which (a) encapsulates information of arbitrary size, type, format, and organization, including other, nested, containers, (b) cryptographically protects that information from all unauthorized Access and Use, (c) provides encrypted storage management   |

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction  |
|--|----------------------|-------------------------|---|
|  |                      |                         | <p>functions for that information, such as hiding the physical storage location(s) of its protected contents, (d) permits the association of itself or its contents with <b>Controls</b> and control information governing <b>(Controlling) Access</b> to and <b>Use</b> thereof, and (e) prevents such <b>Use</b> or <b>Access</b> (as opposed to merely preventing decryption) until it is "opened."</p> <p>(2) A <b>Secure Container</b> can be opened only as expressly <i>Allowed</i> by the associated <b>VDE Control(s)</b>, only within a <i>Secure Processing Environment</i>, and only through decryption of its encrypted header.</p> <p>(3) A <b>Secure Container</b> is not directly accessible to any non-VDE or user calling process. All such calls are intercepted by VDE.</p> <p>(4) The creator of a <b>Secure Container</b> can assign (or allow others to assign) control information to any arbitrary portion of a <b>Secure Container's</b> contents, or to an empty <b>Secure Container</b> (to govern <b>(Control)</b> the later addition of contents to the container, and <b>Access</b> to or <b>Use</b> of those contents).</p> <p>(5) A container is not a <b>Secure Container</b> merely because its contents are encrypted and signed. A <b>Secure Container</b> is itself <b>Secure</b>.</p> <p>(6) All VDE-protected information (including protected content, information about content usage, content-control information, <b>Controls</b>, and <i>Load Modules</i>) is encapsulated within a <b>Secure Container</b> whenever stored outside a <i>Secure Processing Environment</i> or secure database.</p> |

|     | Claim<br>Term/Phrase                   | InterTrust Construction  | Microsoft Construction   |
|-----|--|--|--|
|     |  |  | For the purposes of the construction of "Secure Container," "Secure Processing Environment" and "Load Module" are defined as set forth in item #6, above, and "Access" and "Allow" are defined as set forth in item #8, above.   |
| 21. | tamper resistance<br><br>721.1         | Making tampering more difficult and/or allowing detection of tampering.<br><br>In this definition, "tampering" means using (e.g., observing or altering) in any unauthorized manner, or interfering with authorized use. | <u>tamper resistance</u> : The ability of a <b>Tamper Resistant Barrier</b> to prevent <i>Access</i> , observation, and interference with information or processing encapsulated by the barrier.<br><br>For the purposes of the construction of "Tamper Resistance," " <i>Tamper/Tampering</i> " is defined as: Using (e.g., observing or altering) in any unauthorized manner, or interfering with authorized use.<br><br>For the purposes of the construction of "Tamper Resistance," " <i>Access</i> " is defined as set forth in item # 6, above.  |
| 22. | tamper resistant barrier<br><br>721.34 | Hardware and/or software that provides Tamper Resistance.  | (1) An active device that encapsulates and separates a <b>Protected Processing Environment</b> from the rest of the world.<br>(2) It prevents information and processes within the <b>Protected Processing Environment</b> from being observed, interfered with, and leaving except under appropriate conditions ensuring security.<br>(3) It also <b>Controls</b> external access to the encapsulated <b>Secure</b> resources, processes and information.<br>(4) A <b>Tamper Resistant Barrier</b> is capable of destroying protected information in response to <i>Tampering</i> attempts.<br><br>For the purposes of the construction |

|     | Claim<br>Term/Phrase  | InterTrust Construction   | Microsoft Construction  |
|-----|---|---|---|
|     |   |   | of "Tamper Resistant Barrier,"<br>"Tamper/Tampering" is defined as<br>set forth in item #21, above.   |
| 23. | use<br><br>193.19<br>683.2<br>721.1<br>861.58<br>891.1<br>912.8, 912.35   | Normal English: to put into service<br>or apply for a purpose, to employ.   | (1) To use information is to perform<br>some action on it or with it (e.g.,<br>copying, printing, decrypting,<br>encrypting, saving, modifying,<br>observing, or moving, etc.).<br>(2) In VDE, information Use is<br><i>Allowed</i> only through execution of<br>the applicable VDE Control(s) and<br>satisfaction of all requirements<br>imposed by such execution.<br><br>For the purposes of the construction<br>of "Use," " <i>Allowed</i> " is defined as set<br>forth in item #8 above.   |
| 24. | virtual distribution<br>environment<br><br>900.155<br><br>Also as set forth in<br>each "claim as a<br>whole" by<br>Microsoft. | This term is contained in the<br>preamble of the claim and should<br>not be defined, other than as<br>requiring the individual claim<br>elements. The term "virtual<br>distribution environment" should not<br>be read into claims that do not<br>actually recite it.<br><br>Without waiving its position that no<br>separate definition is required, if<br>required to propose such a<br>definition, InterTrust proposes the<br>following: secure, distributed<br>electronic transaction management<br>and rights protection system for<br>controlling the distribution and/or<br>other usage of electronically<br>provided and/or stored information. | <u>VDE/Virtual Distribution<br/>Environment:</u><br><br>(1) <u>Data Security and Commerce<br/>World</u> : InterTrust's February 13,<br>1995, patent application described as<br>its "invention" a <b>Virtual<br/>Distribution Environment</b> ("VDE<br>invention") for securing,<br>administering, and auditing all<br>security and commerce digital<br>information within its multi-node<br>world (community). <b>VDE</b><br>guarantees to all VDE "participants"<br>identified in the patent application<br>that it will limit all <b>Access</b> to and<br><b>Use</b> (i.e., interaction) of such<br>information to authorized activities<br>and amounts, will ensure any<br>requested reporting of and payment<br>for such Use, and will maintain the<br>availability, secrecy, integrity, non-<br>repudiation and authenticity of all<br>such information present at any of<br>its nodes (including protected<br>content, information about content<br>usage, and content <b>Controls</b> .). |



|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction   |
|--|----------------------|-------------------------|--|
|  |                      |                         | <p>VDE is Secure against at least the threats identified in the February 1995, patent application to this availability (no user may delete the information without authorization), secrecy (neither available nor disclosed to unauthorized persons or processes), integrity (neither intentional nor accidental alteration), non-repudiation (neither the receiver can disavow the receipt of a message nor can the sender disavow the origination of that message) and authenticity (asserted characteristics are genuine). VDE further provides and requires the components and capabilities described below. Anything less than or different than this is not VDE or the described "invention."</p> <p>(2) <u>Secure Processing Environment</u>: At each node where VDE-protected information is <i>Accessed, Used</i>, or assigned control information, VDE requires a <i>Secure Processing Environment</i> (as set forth in item #6).</p> <p>(3) <u>VDE Controls</u>: VDE Allows <i>Access</i> to or <i>Use</i> of protected information and processes only through execution of (and satisfaction of the requirements imposed by) <b>VDE Control(s)</b>.</p> <p>(4) <u>VDE Secure Container</u>: See construction of <b>Secure Container</b>.</p> <p>(5) <u>Non-Circumventable</u>: VDE is non-circumventable (sequestered). It intercepts all attempts by any and all users, processes, and devices, to <i>Access</i> or <i>Use</i>, such as observing, interfering with, or removing) protected information, and prevents all such attempts other than as</p> |

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction   |
|--|----------------------|-------------------------|--|
|  |                      |                         | <p>allowed by execution of (and satisfaction of all requirements imposed by) associated <b>VDE Controls</b> within <i>Secure Processing Environment(s)</i>.</p> <p>(6) <u>Peer to Peer</u>: VDE is peer-to-peer. Each VDE node has the innate ability to perform any role identified in the patent application (e.g., end user, content packager, distributor, <b>Clearinghouse</b>, etc.), and can protect information flowing in any direction between any nodes. VDE is not client-server. It does not pre-designate and restrict one or more nodes to act solely as a "server" (a provider of information (e.g., authored content, control information, etc.) to other nodes) or "client" (a requestor of such information). All types of protected-content transactions can proceed without requiring interaction with any server.</p> <p>(7) <u>Comprehensive Range of Functions</u>: VDE comprehensively governs (<b>Controls</b>) all security and commerce activities identified in the patent application, including (a) metering, budgeting, monitoring, reporting, and auditing information usage, (b) billing and paying for information usage, and (c) negotiating, signing and enforcing contracts that establish users' rights to <i>Access</i> or <i>Use</i> information.</p> <p>(8) <u>User-Configurable</u>: The specific protections governing (<b>Controlling</b>) specific VDE-protected information are specified, modified, and negotiated by VDE's users. For example, VDE enables a consumer to place limits on the nature of content that may be <i>Accessed</i> at her</p> |

|     | Claim<br>Term/Phrase   | InterTrust Construction  | Microsoft Construction   |
|-----|--|--|--|
|     |  |  | <p>node (e.g., no R-rated material) or the amount of money she can spend on viewing certain content, both subject only to other users' senior Controls.</p> <p>(9) <u>General Purpose; Universal</u>: VDE is universal as opposed to being limited to or requiring any particular type of appliance, information, or commerce model. It is a single, unified standard and environment within which an unlimited range of electronic rights protection, data security, electronic currency, and banking applications can run.</p> <p>(10) <u>Flexible</u>: VDE is more flexible than traditional information security and commerce systems. For example, VDE allows consumers to pay for only the user-defined portion of information that the user actually uses, and to pay only in proportion to any quantifiable VDE event (e.g., for only the number of paragraphs displayed from a book), and allows editing the content in VDE containers while maintaining its security.</p> <p>For the purposes of the construction of "VDE," a "<i>Secure Processing Environment</i>" is defined as set forth in item #6, above.</p> <p>For the purposes of the construction of "VDE," "<i>Access</i>" is defined as set forth in item #8, above.</p> |
| 25. | 193.1: "a budget specifying the number of copies which can be made of said digital file" | Normal English, incorporating the separately defined terms: a Budget stating the number of copies that can be made of the digital file referred to earlier in the claim. | A Budget explicitly stating the total number of copies (whether or not decrypted, long-lived, or accessible) that (since creation of the Budget) are authorized to be made of the <i>Digital File</i> by any and all users, devices, and processes. No process,  |

|     | Claim Term/Phrase   | InterTrust Construction  | Microsoft Construction  |
|-----|---|--|---|
|     |   |  | <p>user, or device is able to make another copy of the <i>Digital File</i> once this number of copies has been made.</p> <p>For the purposes of the construction of this phrase, "<i>Digital File</i>" is defined as set forth in item #6, above.</p>   |
| 26. | 193.1: "controlling the copies made of said digital file"   | The nature of this operation is further defined in later claim elements. In context, the copy control determines the conditions under which a digital file may be Copied and the copied file stored on a second device.  | <p><b>Controlling</b> Uses of and <i>Accesses</i> to all copies of the <i>Digital File</i>, by all users, processes, and devices, by executing each of the recited "at least one" <b>Copy Control(s)</b> within <b>VDE Secure Processing Environment(s)</b>. Each <b>Control</b> governs (<b>Controls</b>) only one action, which action may or may not differ among the different "at least one" <b>Controls</b>. All <i>Uses</i> and <i>Accesses</i> are prohibited and incapable of occurring except to the extent <i>Allowed</i> by the "at least one" <b>Copy Control(s)</b>.</p> <p>For the purposes of the construction of this phrase, a "<i>Secure Processing Environment</i>" is defined as set forth in item #6, above, and "<i>Access</i>" and "<i>Allowed</i>" are defined as set forth in item #8, above.</p> |
| 27. | 721.1: "digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance" | Normal English, incorporating the separately defined terms: generating a Digital Signature for the second load module, the Digital Signature Designating that the second load module is for use by a second Device Class. This element further requires that the second Device Class have a different Tamper Resistance or security level than the first Device Class. | <p>(1) <b>Digitally Signing</b> a different ("second") <i>Load Module</i> by using a different ("second") <b>Digital Signature</b> as the signature <i>Key</i>, which signing indicates to any and all devices in the second <b>Device Class</b> that the signor authorized and restricted this <i>Load Module</i> for Use by that device.</p> <p>(2) No VDE device can perform any execution of any <i>Load Module</i> without such authorization. The method ensures that the <i>Load Module</i> cannot execute in a particular <b>Device Class</b> and ensures</p>   |

|     | Claim Term/Phrase  | InterTrust Construction   | Microsoft Construction  |
|-----|--|---|---|
|     | and security level different from the at least one of tamper resistance and security level of the first device class”  |   | <p>that no device in that <b>Device Class</b> has the <i>Key(s)</i> necessary to verify the <b>Digital Signature</b>.</p> <p>(3) All devices in the first <b>Device Class</b> have the same persistent (not just occasional) and identified level of <b>Tamper Resistance</b> and the same persistent and identified level of security. All devices in the second <b>Device Class</b> have the same persistent and identified level of <b>Tamper Resistance</b> and same persistent and identified level of security.</p> <p>(4) The identified level of <b>Tamper Resistance</b> or identified level of security (or both) for the first <b>Device Class</b>, is greater than or less than the identified level of <b>Tamper Resistance</b> or identified level of security for the second <b>Device Class</b>.</p> <p>For the purposes of the construction of this phrase, a “<i>Load Module</i>” is defined as set forth in item #6, above, and “<i>Key</i>” is defined as set forth in item #14, above.</p> |
| 28. | 891.1: “securely applying, at said first appliance through use of said at least one resource said first entity’s control and said second entity’s control to govern use of said data item” | Normal English, incorporating the separately defined terms: the first entity’s Control and the second entity’s Control are Securely applied to govern Use of the data item, the act of Securely applying involving use of the resource. | <p>(1) Processing the resource (component part of a first appliance’s <b>Secure Operating Environment</b>) within the <b>Secure Operating Environment</b>’s special-purpose <b>Secure Processing Unit (SPU)</b> to execute the first <b>Control</b> and second <b>Control</b> in combination within the <b>SPU</b>.</p> <p>(2) This execution of these <b>Controls</b> governs (<b>Controls</b>) all <b>Use</b> of the data item by all users, processes, and devices.</p> <p>(3) The processing of the resource and execution of the <b>Controls</b> cannot be observed from outside the <b>SPU</b> and is performed only after the</p>  |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|     | Claim Term/Phrase   | InterTrust Construction  | Microsoft Construction  |
|-----|---|--|---|
|     |   |  | <p>integrity of the resource and <b>Controls</b> is cryptographically verified.</p> <p>(4) A Secure Processing Unit is a special-purpose unit isolated from the rest of the world in which a hardware <b>Tamper Resistant Barrier</b> encapsulates a processor and internal <b>Secure</b> memory.</p> <p>(5) The processor cryptographically verifies the integrity of all code loaded from the <b>Secure</b> memory prior to execution, executes only the code that the processor has authenticated for its <b>Use</b>, and is otherwise <b>Secure</b>.</p>  |
| 29. | 900.155: "derives information from one or more aspects of said host processing environment"                         | Normal English, incorporating the separately defined terms: Derives (including creates) information based on at least one Aspect of the previously referred to Host Processing Environment                 | <p>(1) <b>Deriving</b> from the <b>Host Processing Environment</b> hardware one or more values that uniquely and persistently identify the <b>Host Processing Environment</b> and distinguish it from other <b>Host Processing Environments</b>.</p> <p>(2) The "one or more aspects of said host processing environment" are persistent elements or properties of the <b>Host Processing Environment</b> itself that are capable of being used to distinguish it from other environments, as opposed to, e.g., data or programs stored within the mass storage or main memory, or processes executing within the <b>Host Processing Environment</b>.</p> |
| 30. | 912.8: "identifying at least one aspect of an execution space required for use and/or execution of the load module" | Normal English, incorporating the separately defined terms: identifying an Aspect (e.g. security level) of an execution space that is needed in order for the load module to execute or otherwise be used. | <p>(1) Defining fully, without reference to any other information, at least one of the persistent elements or properties (<b>Aspects</b>) (that are capable of being used to distinguish it from other environments of an execution space) that are required for any <b>Use</b>, and/or for any execution, of the <i>Load Module</i>.</p> <p>(2) An execution space without all of those required aspects is</p>  |

EXHIBIT B TO JOINT CLAIM CONSTRUCTION STATEMENT

|  | Claim<br>Term/Phrase | InterTrust Construction | Microsoft Construction   |
|--|----------------------|-------------------------|--|
|  |                      |                         | <p>incapable of making any such execution and/or other Use (e.g., <b>Copying</b>, displaying, printing) of the <i>Load Module</i>.</p> <p>For the purposes of the construction of this phrase, a "<i>Load Module</i>" is defined as set forth in item #6, above.</p> |

## **EXHIBIT C**

### **PLR 4-3(b) – Identification of Supporting Evidence**

The following represents InterTrust's list of evidence relevant to construction of the disputed terms and phrases.

Notes:

1. InterTrust reserves the right to supplement this list as needed to respond to changed constructions proffered by Microsoft. InterTrust also reserves the right to rely on evidence cited in the original version of this Exhibit, filed February 3, 2003.
2. In the following list, certain terms and phrases include other, separately defined terms. In such cases, the evidence supporting the separately defined term is also relevant to construction of the larger term.
3. The InterTrust patents include overlapping specifications, in which the same text may be found in two or more specifications. Where only one of the specifications is cited, InterTrust reserves the right to substitute citations for the same text in the other specifications.
4. Highlighting has been used to indicate added emphasis.
5. Each claim term is followed by a list of all patent claims in which the term appears (e.g., "193.15" means claim 15 from the '193 patent).

Key to abbreviations:

USP = United States Patent  
'193 patent = USP 6,253,193  
'683 patent = USP 6,185,683  
'721 patent = USP 6,157,721  
'891 patent = USP 5,982,891  
'861 patent = USP 5,920,861  
'912 patent = USP 5,917,912  
'900 patent = USP 5,892,900



|    | Claim Term / Phrase                                | InterTrust Evidence   |
|----|--|---|
| 1. | <p>aspect</p> <p>683.2, 861.58, 900.155, 912.8</p> | <p><u>Patent Specifications</u></p> <p>1(A)</p> <p>This reinitialization mechanism would permit CPU/SPU 2650 to be initialized several times, facilitating testing and/or re-use for different applications, while protecting all security-relevant aspects of its operation.</p> <p>'900 patent at 77:15-19.</p> <hr/> <p>1(B)</p> <p>In addition, the overall software-based tamper resistant barrier 674 and associated PPE system is sufficiently complex so that it is difficult to tamper with a part of it without destroying other aspects of its functionality (i.e., a "defense in depth").</p> <p>'900 patent at 236:3-7.</p> <hr/> <p>1(C)</p> <p>As with any system incorporating "applications" and "operating systems," the boundary between these aspects of an overall system can be ambiguous.</p> <p>'193 patent at 83:30-32.</p> <hr/> <p>1(D)</p> <p>Since SPE 503 in the preferred embodiment runs within the confines of an SPU 500, one aspect of this device driver 736 is to provide low level communications services with the SPU 500 hardware.</p> <p>'193 patent at 95:27-30.</p> <hr/> <p>1(E)</p> <p>Templates may present one or more models that describe various</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="609 331 1430 474"><del>aspects</del> of a content object and how the object should be created including employing secure atomic methods that are used to create, alter, and/or destroy permissions records 808 and/or associated budgets, etc.</p> <p data-bbox="521 510 829 541">'193 patent at 260:42-47.</p> <hr data-bbox="516 577 1451 583"/> <p data-bbox="521 625 574 657"><b>1(F)</b></p> <p data-bbox="609 693 1422 829">In accordance with one <del>aspect</del> of how to advantageously use descriptive data structures in accordance with a preferred embodiment of this invention, a machine readable descriptive data structure may be created by a provider to describe the layout of the</p> <p data-bbox="609 936 1398 999">provider's particular rights management data structure(s) such as secure containers.</p> <p data-bbox="521 1041 797 1073">'861 patent at 6:24-29.</p> <hr data-bbox="516 1108 1451 1115"/> <p data-bbox="521 1157 574 1188"><b>1(G)</b></p> <p data-bbox="609 1224 1409 1325">Controls 316 may provide rules and associated consequences for controlling or otherwise affecting the use or other <del>aspects</del> of what value chain participant 602 can do with DDS 200.</p> <p data-bbox="521 1360 781 1392">'861 patent at 17:3-6.</p> |

|    | Claim Term /<br>Phrase       | InterTrust Evidence  |
|----|------------------------------|--|
| 2. | authentication<br><br>193.15 | <p><u>Patent Specifications</u></p> <p>2(A)</p> <p>To increase the security of security barrier 502 even further, it is possible to encase or include SPU 500 in one or more further physical enclosures such as, for example: epoxy or other "potting compound"; further module enclosures including additional self-destruct, self-disabling or other features activated when tampering is detected; further modules providing additional security protections such as requiring password or other authentication to operate; and the like.</p> <p>'193 patent at 64:29-37.</p> <hr/> <p>2(B)</p> <p>It may also or alternatively provide or include one or more passwords or other information used to identify or otherwise verify/authenticate an individual's identity, such as voice print and retinal scan information.</p> <p>'193 patent at 236:21-25.</p> <hr/> <p>2(C)</p> <p>This certification process in the preferred embodiment may be used to permit a VDE electronic appliance to present one or more "certificates" authenticating that it (or its key) can be trusted. As described above, this "certification" process may be used by one PPE 650 to "certify" that it is an authentic VDE PPE, it has a certain level of security and capability set (e.g., it is hardware based rather than merely software based), etc. Briefly, the "certification" process may involve using a certificate private key of a certification key pair to encrypt a message including another VDE node's public-key. The private key of a certification key pair is preferably used to generate a PPE certificate. It is used to encrypt a public-key of the PPE. A PPE certificate can either be stored in the PPE, or it may be stored in a certification repository.</p> <p>'193 patent at 213:1-15.</p> |

| Claim Term / Phrase      | InterTrust Evidence  |           |             |                      |  |             |  |                          |   |
|--------------------------|--|-----------|-------------|----------------------|--|-------------|--|--------------------------|---|
|                          | <p>2(D)</p> <p>SPE Authentication Manager/Service Communications Manager 564</p> <p>The <u>Authentication Manager</u>/Service Communications Manager 564 supports calls for user password validation and “ticket” generation and validation. It may also support secure communications between SPE 503 and an external node or device (e.g., a VDE administrator or distributor). It may support the following examples of authentication-related service requests in the preferred embodiment:</p> <table><tr><th>Call Name</th><th>Description</th></tr><tr><td colspan="2"><u>User Services</u></td></tr><tr><td>Create User</td><td>Creates a new user and stores Name Services Records (NSRs) for use by the Name Services Manager 752.</td></tr><tr><td><u>Authenticate User</u></td><td>Authenticates a user for use of the system. This request lets the caller authenticate as a <u>specific user ID</u>. <u>Group membership is also authenticated</u> by this request. The authentication returns a “ticket” for the user.</td></tr></table> <p>‘193 patent at 123:21-42.</p> | Call Name | Description | <u>User Services</u> |  | Create User | Creates a new user and stores Name Services Records (NSRs) for use by the Name Services Manager 752. | <u>Authenticate User</u> | Authenticates a user for use of the system. This request lets the caller authenticate as a <u>specific user ID</u> . <u>Group membership is also authenticated</u> by this request. The authentication returns a “ticket” for the user. |
| Call Name                | Description  |           |             |                      |  |             |  |                          |   |
| <u>User Services</u>     |  |           |             |                      |  |             |  |                          |   |
| Create User              | Creates a new user and stores Name Services Records (NSRs) for use by the Name Services Manager 752.   |           |             |                      |  |             |  |                          |   |
| <u>Authenticate User</u> | Authenticates a user for use of the system. This request lets the caller authenticate as a <u>specific user ID</u> . <u>Group membership is also authenticated</u> by this request. The authentication returns a “ticket” for the user.  |           |             |                      |  |             |  |                          |   |

|    | Claim Term / Phrase | InterTrust Evidence   |
|----|---------------------|---|
| 3. | budget<br><br>193.1 | <p><u>Patent Specifications</u></p> <p>3(A)</p> <p>PERC 808 may also contain or refer to <u>budgets containing potentially valuable quantities/values</u>. Such budgets may be stored within a traveling object itself, or they may be delivered separately and protected by highly secure communications keys and administrative object keys and management database techniques.</p> <p>'193 patent at 132:60-65.</p> <hr/> <p>3(B)</p> <p><u>User Data Elements (UDEs) 1200 and Method Data Elements (MDEs) 1202 in the preferred embodiment store data.</u> There are many types of UDEs 1200 and MDEs 1202 provided by the preferred embodiment. In the preferred embodiment, each of these different types of data structures shares a common overall format including a common header definition and naming scheme. Other UDEs 1200 that share this common structure include "local name services records" (to be explained shortly) and account information for connecting to other VDE participants. These elements are not necessarily associated with an individual user, and may therefore be considered MDEs 1202. All UDEs 1200 and all MDEs 1202 provided by the preferred embodiment may, if desired, (as shown in Figure 16) be stored in a common physical table within secure database 610, and database access processes may commonly be used to access all of these different types of data structures.</p> <p>In the preferred embodiment, PERCs 808 and user rights table records are types of UDE 1200. <u>There are many other types of UDEs 1200/MDEs 1202</u>, including for example, meters, meter trails, <u>budgets</u>, budget trails, and audit trails.</p> <p>'193 patent at 142:41-61.</p> <hr/> <p>3(C)</p> <p>In the example shown in Figure 41d, a distributor at a VDE distributor node (106) might <u>request budget</u> from a content creator at another node (102). This request may be made in the context of a secure VDE communication or it may be passed in an "out-of-</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>channel" communication (e.g. a telephone call or letter). The creator 102 may decide to grant budget to the distributor 106 and processes a distribute event (1452 in BUDGET method 1510 at VDE node 102). A result of processing the distribute event within the BUDGET method might be a secure communication (1454) between VDE nodes 102 and 106 by which a budget granting use and redistribute rights to the distributor 106 may be transferred from the creator 102 to the distributor. The distributor's VDE node 106 may respond to the receipt of the budget information by processing the communication using the reply process 1475B of the BUDGET method 1510. The reply event processing 1475B might, for example, install a budget and PERC 808 within the distributor's VDE 106 node to permit the distributor to access content or processes for which access is control at least in part by the budget and/or PERC. At some point, the distributor 106 may also desire to use the content to which she has been granted rights to access.</p> <p>After registering to use the content object, the user 112 would be required to utilize an array of "use" processes 1476C to, for example, open, read, write, and/or close the content object as part of the use process.</p> <p>Once the distributor 106 has used some or all of her budget, she may desire to obtain additional budget. The distributor 106 might then initiate a process using the BUDGET method request process (1480B). Request process 1480B might initiate a communication (1482AB) with the content creator VDE node 102 requesting more budget and perhaps providing details of the use activity to date (e.g., audit trails). The content creator 102 processes the 'get more budget' request event 1482AB using the response process (1484A) within the creator's BUDGET method 1510A. Response process 1484A might, for example, make a determination if the use information indicates proper use of the content, and/or if the distributor is credit worthy for more budget. The BUDGET method response process 1484A might also initiate a financial transaction to transfer funds from the distributor to pay for said use, or use the distribute process 1472A to distribute budget to the distributor 106. A response to the distributor 106 granting more budget (or denying more budget) might be sent immediately as a response to the request communication 1482AB, or it might be sent at a later time as part of a separate communication. The response communication, upon being received at the distributor's VDE node 106, might be processed using the reply process 1475B within the distributor's copy of the BUDGET method 1510B. The reply process 1475B might then process the additional budget in the same manner as described above.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>The chain of handling and control may, in addition to posting <u>budget</u> information, also pass control information that governs the manner in which said <u>budget</u> may be utilized. For example, the control information specified in the above example may also contain control information describing the process and limits that apply to the distributor's redistribution of the right to use the creator's content object. Thus, when the distributor responds to a <u>budget</u> request from a user (a communication between a user at VDE node 112 to the distributor at VDE node 106 similar in nature to the one described above between VDE nodes 106 and 102) using the distribute process 1472B within the distributor's copy of the <u>BUDGET method 1510B</u>, a distribution and request/response/reply process similar to the one described above might be initiated.</p> <p>'193 patent at 172:61-174:29.</p> <hr/> <p><b>3(D)</b></p> <p>BILLING method 406 may then pass the event on to a BUDGET method 408. BUDGET method 408 sets limits and records transactional information associated with those limits. For example, <u>BUDGET method 408 may store budget information in a budget UDE</u>, and may store an audit record in a budget trail UDE. BUDGET method 408 may result in a "budget remaining" field in a budget UDE being decremented by an amount specified by BILLING method 406.</p> <p>'193 patent at 182:22-30.</p> <hr/> <p><b>3(E)</b></p> <p><u>BUDGET method 1510</u> may read and update <u>budget information</u> within a BUDGET method UDE,</p> <p>'193 patent at 184:67-185:1.</p> <hr/> <p><b>3(F)</b></p> <p>Figure 5A shows how the virtual distribution environment 100, in a <u>preferred embodiment</u>, may package information elements (content) into a "container" 302 so the information can't be accessed except as</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>provided by its “rules and controls.” Normally, the container 302 is electronic rather than physical. Electronic container 302 in one example comprises “digital” information having a well defined structure. Container 302 and its contents can be called an “object 300.”</p> <p>The Figure 5A example shows items “within” and enclosed by container 302. However, container 302 may “contain” items without those items actually being stored within the container. For example, the container 302 may reference items that are available elsewhere such as in other containers at remote sites. Container 302 may reference items available at different times or only during limited times. Some items may be too large to store within container 302. Items may, for example, be delivered to the user in the form of a “live feed” of video at a certain time. Even then, the container 302 “contains” the live feed (by reference) in this example.</p> <p>Container 302 may contain information content 304 in electronic (such as “digital”) form. Information content 304 could be the text of a novel, a picture, sound such as a musical performance or a reading, a movie or other video, computer software, or just about any other kind of electronic information you can think of. Other types of “objects” 300 (such as “administrative objects”) may contain “administrative” or other information instead of or in addition to information content 304.</p> <p><u>In the Figure 5A example, container 302 may also contain “rules and controls” in the form of:</u></p> <ul style="list-style-type: none"> <li>(a) a “permissions record” 808;</li> <li>(b) “budgets” 308; and</li> <li>(c) “other methods” 1000.</li> </ul> <p><u>Figure 5B gives some additional detail about permissions record 808, budgets 308 and other methods 1000.</u> The “permissions record” 808 specifies the rights associated with the object 300 such as, for example, who can open the container 302, who can use the object's contents, who can distribute the object, and what other control mechanisms must be active. For example, permissions record 808 may specify a user's rights to use, distribute and/or administer the container 302 and its content. Permissions record 808 may also specify requirements to be applied by the budgets 308 and “other methods” 1000. Permissions record 808 may also contain security related information such as scrambling and descrambling “keys.”</p> |



| Claim Term / Phrase           | InterTrust Evidence  |                  |   |             |                    |                       |  |                  |                          |                               |                                |        |   |
|-------------------------------|--|------------------|---|-------------|--------------------|-----------------------|--|------------------|--------------------------|-------------------------------|--------------------------------|--------|---|
|                               | <p><b>Budgets</b> 308 shown in Figure 5B are a special type of <b>method</b> 1000 that may specify, among other things, limitations on usage of information content 304, and how usage will be paid for. <b>Budgets</b> 308 can specify, for example, how much of the total information content 304 can be used and/or copied. The methods 310 may prevent use of more than the amount specified by a specific budget.</p> <p>“Other methods” 1000 define basic operations used by “rules and controls.” Such “methods” 1000 may include, for example, how usage is to be “metered,” if and how content 304 and other information is to be scrambled and descrambled, and other processes associated with handling and controlling information content 304. For example, methods 1000 may record the identity of anyone who opens the electronic container 302, and can also control how information content is to be charged based on “metering.” Methods 1000 may apply to one or several different information contents 304 and associated containers 302, as well as to all or specific portions of information content 304.</p> <p>‘193 patent at 58:38-59:37.</p> <hr/> <p><b>3(G)</b></p> <p>FIGURES 5A and 5B show an <b>example</b> of an “object”;</p> <p>‘193 patent at 50:18.</p> <hr/> <p><b>3(H)</b></p> <table><tr><th>Field type</th><th>Format</th><th>Typical Use</th><th>Description or Use</th></tr><tr><td>Ascending Use Counter</td><td>byte, short, long, or unsigned versions of the same widths</td><td>Meter/<br/>Budget</td><td>Ascending count of uses.</td></tr><tr><td><b>Descending Use Counter</b></td><td>byte, short, long, or unsigned</td><td>Budget</td><td><b>Descending count of permitted use; e.g., remaining</b></td></tr></table> | Field type       | Format  | Typical Use | Description or Use | Ascending Use Counter | byte, short, long, or unsigned versions of the same widths | Meter/<br>Budget | Ascending count of uses. | <b>Descending Use Counter</b> | byte, short, long, or unsigned | Budget | <b>Descending count of permitted use; e.g., remaining</b> |
| Field type                    | Format   | Typical Use      | Description or Use  |             |                    |                       |  |                  |                          |                               |                                |        |   |
| Ascending Use Counter         | byte, short, long, or unsigned versions of the same widths   | Meter/<br>Budget | Ascending count of uses.                                  |             |                    |                       |  |                  |                          |                               |                                |        |   |
| <b>Descending Use Counter</b> | byte, short, long, or unsigned   | Budget           | <b>Descending count of permitted use; e.g., remaining</b> |             |                    |                       |  |                  |                          |                               |                                |        |   |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="824 331 1279 436">versions of<br/>the same<br/>widths <span style="float: right;">budget</span></p> <p data-bbox="526 470 834 512">'193 patent at 143:57-65.</p> <hr/> <p data-bbox="526 583 574 625"><b>3(I)</b></p> <p data-bbox="613 659 1429 798">As with standard VDE objects 300, a user may be required to contact a clearinghouse service to acquire additional budgets if the user wishes to continue to use the traveling object after the exhaustion of an available budget(s) . . . .</p> <p data-bbox="526 831 834 873">'193 patent at 131:10-13.</p> <hr/> <p data-bbox="526 945 574 987"><b>3(J)</b></p> <p data-bbox="613 1012 1451 1843">Once the distributor 106 has used some or all of her budget, she may desire to obtain additional budget. The distributor 106 might then initiate a process using the BUDGET method request process (1480B). Request process 1480B might initiate a communication (1482AB) with the content creator VDE node 102 requesting more budget and perhaps providing details of the use activity to date (e.g., audit trails). The content creator 102 processes the 'get more budget' request event 1482AB using the response process (1484A) within the creator's BUDGET method 1510A. Response process 1484A might, for example, make a determination if the use information indicates proper use of the content, and/or if the distributor is credit worthy for more budget. The BUDGET method response process 1484A might also initiate a financial transaction to transfer funds from the distributor to pay for said use, or use the distribute process 1472A to distribute budget to the distributor 106. A response to the distributor 106 granting more budget (or denying more budget) might be sent immediately as a response to the request communication 1482AB, or it might be sent at a later time as part of a separate communication. The response communication, upon being received at the distributor's VDE node 106, might be processed using the reply process 1475B within the distributor's copy of the BUDGET method 1510B. The reply process 1475B might then process the additional budget in the same manner as described above.</p> <p data-bbox="526 1877 883 1919">'193 patent at 173:21-174:14.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="524 338 586 369"><b>3(K)</b></p> <p data-bbox="613 407 1451 1236">During the same or different communications exchange, the same or different clearinghouse may handle the end user's request for additional budget and/or permission pertaining to VDE object 300. For example, the end user's electronic appliance 600 may (e.g., in response to a user input request to access a particular VDE object 300) send an administrative object to the clearinghouse requesting budgets and/or other permissions allowing access (Block 1164). As mentioned above, such requests may be transmitted in the form of one or more administrative objects, such as, for example, a single administrative object having multiple "events" associated with multiple requested budgets and/or other permissions for the same or different VDE objects 300. The clearinghouse may upon receipt of such a request, check the end user's credit, financial records, business agreements and/or audit histories to determine whether the requested budgets and/or permissions should be given. The clearinghouse may, based on this analysis, send one or more responsive administrative objects which cause the end user's electronic appliance 600 to update its secure database in response (Block 1166, 1168). This updating might, for example, comprise replacing an expired PERC 808 with a fresh one, modifying a PERC to provide additional (or lesser) rights, etc. Steps 1164-1168 may be repeated multiple times in the same or different communications session to provide further updates to the end user's secure database 610.</p> <p data-bbox="524 1276 829 1308">'193 patent at 162:39-65.</p> <hr data-bbox="516 1346 1459 1352"/> <p data-bbox="521 1392 748 1423"><b><u>Extrinsic Sources</u></b></p> <p data-bbox="521 1461 578 1493"><b>3(L)</b></p> <p data-bbox="610 1530 1446 1942"><b>budget</b> <i>n.</i> <b>1.a.</b> An itemized summary of estimated or intended expenditures for a given period along with proposals for financing them: <i>submitted the annual budget to Congress.</i> <b>b.</b> A systematic plan for the expenditure of a usually fixed resource, such as money or time, during a given period: <i>A new car will not be part of our budget this year.</i> <b>c.</b> The total sum of money allocated for a particular purpose or period of time: <i>a project with an annual budget of five million dollars.</i> <b>2.</b> <u>A stock or collection with definite limits</u>: <i>"his budget of general knowledge."</i> (William Hazlitt). – <b>budget</b> <i>v.</i> —et-ed, et-ing, -ets. —tr. <b>1.</b> To plan in advance the expenditure of: <i>needed help budgeting our income; budgeted my time wisely.</i> <b>2.</b> To enter or account for in a budget: <i>forgot to budget</i></p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p><i>the car payments. –intr. To make or use a budget. –budget adj. 1. Of or relating to a budget: budget items approved by Congress. 2. Appropriate to a budget; inexpensive: a budget car; budget meals.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 249.</p> |

|    | Claim Term / Phrase         | InterTrust Evidence  |
|----|-----------------------------|--|
| 4. | clearinghouse<br><br>193.19 | <p><u>Patent Specifications</u></p> <p>4(A)</p> <p>Clearinghouses may provide independent <u>financial services</u>, such as credit and/or billing services, and can serve as <u>distributors and/or creators</u>.</p> <p>'193 patent at 267:40-42.</p> <hr/> <p>4(B)</p> <p>if appropriate credit (e.g. an electronic clearinghouse account from a <u>clearinghouse such as VISA or AT&amp;T</u>) is available.</p> <p>'193 patent at 25:22-24.</p> <hr/> <p>4(C)</p> <p>clearinghouses that gather usage information regarding, and bill for the use of, electronic information.</p> <p>'193 patent at 3:32-33.</p> <hr/> <p>4(D)</p> <p>in certain models, <u>a clearinghouse might also serve as a rights distribution agent</u> who provides one or more rights to certain value chain participants, which one or more rights may be "attached" to one or more rights to use the clearinghouse's credit (if said clearinghouse is, at least in part, a <u>financial clearinghouse</u> (such a control information provider may alternatively, or in addition, restrict other users' rights.</p> <p>'193 patent at 269:59-65.</p> <hr/> <p>4(E)</p> <p>A document may have an attribute requiring that each use of the document be reported to a central <u>document tracking clearinghouse</u>. This could be used by the organization to track specific documents,</p> |

| Claim Term / Phrase                         | InterTrust Evidence  |      |             |   |  |                 |   |
|---|--|------|-------------|---|--|-----------------|---|
|   | <p>to identify documents used by any particular user and/or group of users to track documents with specific attributes (e.g., sensitivity), etc.</p> <p>'193 patent at 280:18-24.</p> <hr/> <p><b>4(F)</b></p> <p>In this Figure 2 example, information relating to content use is, as shown by arrow 114, reported to a <u>financial clearinghouse</u> 116. Based on this "reporting," the financial clearinghouse 116 may generate a bill and send it to the content user 112 over a "reports and payments" network 118. Arrow 120 shows the content user 112 providing payments for content usage to the financial clearinghouse 116. Based on the reports and payments it receives, the financial clearinghouse 116 may provide reports and/or payments to the distributor 106.</p> <p>'193 patent at 55:57-66.</p> <hr/> <p><b>4(G)</b></p> <p>The "<u>financial clearinghouse</u>" 116 shown in Figure 2 may also be a "VDE administrator." Financial clearinghouse 116 in its VDE administrator role sends "administrative" information to the VDE participants. This administrative information helps to keep the virtual distribution environment 100 operating properly. The "VDE administrator" and financial clearinghouse roles may be performed by different people or companies, and there can be more than one of each.</p> <p>'193 patent at 56:16-24.</p> <hr/> <p><b>4(H)</b></p> <p>A summary of the roles of the various participants of virtual distribution environment 100 is set forth in the table below:</p> <table> <tr> <th>Role</th> <th>Description</th> </tr> <tr> <td colspan="2"><u>"Traditional"</u><br/><u>Participants</u></td> </tr> <tr> <td>Content creator</td> <td>Packager and initial distributor of digital</td> </tr> </table> | Role | Description | <u>"Traditional"</u><br><u>Participants</u> |  | Content creator | Packager and initial distributor of digital |
| Role  | Description  |      |             |   |  |                 |   |
| <u>"Traditional"</u><br><u>Participants</u> |  |      |             |   |  |                 |   |
| Content creator                             | Packager and initial distributor of digital  |      |             |   |  |                 |   |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>information</p> <p>Content Owner      Owner of the digital information.</p> <p>Distributors      Provide rights distribution services for budgets and/or content.</p> <p>Auditor      Provides services for processing and reducing usage based audit trails.</p> <p>Clearinghouse      Provides intermediate store and forward services for content and audit information. Also, typically provides a platform for other services, including third party financial providers and auditors.</p> <p>'193 patent at 255:33-51.</p> <hr/> <p><b>4(I)</b></p> <p>Further Chain of Handling Model</p> <p>As described in connection with Figure 2, there are four (4) "participant" instances of VDE 100 in one example of a VDE chain of handling and control used, for example, for content distribution.</p> <p>'193 patent at 253:64-254:1.</p> <hr/> <p><b>4(J)</b></p> <p>FIGURE 2 illustrates an example of a chain of handling and control;</p> <p>'193 patent at 50:8-9.</p> <hr/> <p><b>4(K)</b></p> <p>a "trusted" financial clearinghouse (e.g., VISA, Mastercard).</p> <p>'193 patent at 41:8-9.</p> |

|    | Claim Term / Phrase     | InterTrust Evidence   |
|----|-------------------------|---|
| 5. | compares<br><br>900.155 | <p><u>Patent Specifications</u></p> <p>5(A)</p> <p>Comparing Figure 50 with Figure 49 reveals that the same overall high level processing may typically be performed for READ method 1650 as was described in connection with OPEN method 1500.</p> <p>'900 patent at 195:9-12.</p> <hr/> <p>5(B)</p> <p>As compared to Figure 2, Figure 77 includes a new "client administrator" participant 700.</p> <p>'900 patent at 280:63-65.</p> <hr/> <p>5(C)</p> <p>VDE content, and the electronic agreements associated with said content, can be employed and progressively manipulated in commercial ways which reflect traditional business practices for non-electronic products (though VDE supports greater flexibility and efficiency compared with most of such traditional models).</p> <p>'900 patent at 322:15-20.</p> <hr/> <p><u>Extrinsic Sources</u></p> <p>5(D)</p> <p><b>compare</b> <i>v. tr.</i> 1. To consider or describe as similar, equal, or analogous; liken. 2. <i>Abbr. cp.</i> To examine in order to note the similarities or differences of. 3. <i>Grammar.</i> To form the positive, comparative, or superlative degree of (an adjective or adverb). – <i>intr.</i> 1. To be worthy of comparison; bear comparison: <i>two concert halls that just do not compare.</i> 2. To draw comparisons.</p> <p><b>comparison</b> <i>n.</i> 1.a. The act of comparing or the process of being compared.</p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 384.</p> |



|    | Claim Term /<br>Phrase                     | InterTrust Evidence   |
|----|--|---|
| 6. | component<br>assembly<br><br>912.8, 912.35 | <p><u>Patent Specifications</u></p> <p>6(A)</p> <p>ROS VDE functions 604 may be based on segmented, independently loadable executable "component assemblies" 690. These component assemblies 690 are independently securely deliverable. <u>The component assemblies 690 provided by the preferred embodiment comprise code and data elements that are themselves independently deliverable.</u> Thus, each component assembly 690 provided by the preferred embodiment is comprised of independently securely deliverable elements which may be communicated using VDE secure communication techniques, between VDE secure subsystems.</p> <p>These component assemblies 690 are the basic functional unit provided by ROS 602. <u>The component assemblies 690 are executed to perform operating system or application tasks.</u></p> <p>'193 patent at 83:12-26.</p> <hr/> <p>6(B)</p> <p><u>Components 690 are preferably designed to be easily separable and individually loadable. ROS 602 assembles these elements together into an executable component assembly 690 prior to loading and executing the component assembly (e.g., in a secure operating environment such as SPE 503 and/or HPE 655).</u></p> <p>'193 patent at 83:43-48.</p> <hr/> <p>6(C)</p> <p><u>Thus, the channel 594 is the mechanism in the preferred embodiment that collects together or assembles the elements shown in Figure 11E into a component assembly 690 that may be used for event processing.</u></p> <p>'193 patent at 115:67-116:4.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="526 331 586 363"><b>6(D)</b></p> <p data-bbox="618 401 1365 474">In the preferred embodiment, ROS 602 assembles component assemblies 690 based on the following types of elements:</p> <p data-bbox="618 510 1365 716">Permissions Records ("PERC"s) 808;<br/> Method "Cores" 1000;<br/> Load Modules 1100;<br/> Data Elements (e.g., User Data Elements ("UDEs") 1200 and Method Data Elements ("MDEs") 1202); and<br/> Other component assemblies 690.</p> <p data-bbox="526 751 813 783">'193 patent at 85:21-29.</p> <hr/> <p data-bbox="526 867 581 898"><b>6(E)</b></p> <p data-bbox="618 936 1406 1104">The selected method event record 1012, in turn, specifies the appropriate information (e.g., load module(s) 1100, data element UDE(s) and MDE(s) 1200, 1202, and/or PERC(s) 808) used to construct a component assembly 690 for execution in response to the event that has occurred.</p> <p data-bbox="526 1140 824 1171">'193 patent at 138:31-36.</p> <hr/> <p data-bbox="526 1255 574 1287"><b>6(F)</b></p> <p data-bbox="618 1325 1365 1423">The reciprocal process 1454 may be based on a component assembly 690 (e.g., one or more load modules 1100, data, and optionally other methods present in the VDE node 600B).</p> <p data-bbox="526 1459 824 1491">'193 patent at 171:39-42.</p> <hr/> <p data-bbox="526 1575 574 1606"><b>6(G)</b></p> <p data-bbox="618 1644 1430 1745">One important security layer involves ensuring that certain component assemblies 690 are formed, loaded and executed only in secure execution space such as provided within an SPU 500.</p> <p data-bbox="526 1780 808 1812">'193 patent at 87:35-38.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="526 338 586 369"><b>6(H)</b></p> <p data-bbox="613 407 1455 716">ROS 602 provided by the preferred embodiment responds to an event by specifying and beginning processes to process the event. These processes are, in the preferred embodiment, based on methods 1000. Since there are an unlimited number of different types of events, the preferred embodiment supports an unlimited number of different processes to process events. This flexibility is supported by the dynamic creation of component assemblies 690 from independently deliverable modules such as method cores 1000', load modules 1100, and data structures such as UDEs 1200.</p> <p data-bbox="526 758 870 789">'193 patent at 169:62-170:4.</p> <hr data-bbox="516 831 1455 842"/> <p data-bbox="526 873 570 905"><b>6(I)</b></p> <p data-bbox="613 942 1438 1041">In the preferred embodiment, ROS 602 assembles securely independently deliverable elements into a component assembly 690 based in part on context parameters (e.g., object, user).</p> <p data-bbox="526 1083 813 1115">'193 patent at 84:17-20.</p> <hr data-bbox="516 1146 1455 1157"/> <p data-bbox="526 1188 570 1220"><b>6(J)</b></p> <p data-bbox="613 1262 1422 1640">This "channel 0" "open channel" task may then issue a series of requests to secure database manager 566 to obtain the "blueprint" for constructing one or more component assemblies 690 to be associated with channel 594 (block 1127). In the preferred embodiment, this "blueprint" may comprise a PERC 808 and/or URT 464. . . . The preferred embodiment process may next use the "blueprint" to access (e.g., the secure database manager 566 and/or from load module execution manager library(ies) 568) the appropriate "control method" that may be used to, in effect, supervise execution of all of the other methods 1000 within the channel 594 (block 1131).</p> <p data-bbox="526 1682 1000 1713">'193 patent at 112:46-51, 112:63-113:2.</p> <hr data-bbox="516 1745 1455 1755"/> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p><b><u>File Histories</u></b></p> <p>6(K)</p> <p>Column 1, lines 33-65 [of Fischer 5,748,960] describes “data types” or “classes” in object-oriented programming that meets <u>the term “component” recited in the instant claims (i.e. code and data elements that are independently deliverable).</u></p> <p>‘912 Patent File History, 9/22/98 Office Action, pp. 2-3.</p> |

|    | Claim Term / Phrase                    | InterTrust Evidence  |
|----|--|--|
| 7. | contain<br><br>683.2, 912.8,<br>912.35 | <p><u>Patent Specifications</u></p> <p>7(A)</p> <p>A VDE content container is an object that <b>contains</b> both content (for example, commercially distributed electronic information products such as computer software programs, movies, electronic publications or reference materials, etc.) and certain control information related to the use of the object's content.</p> <p>'193 patent at 19:15-21.</p> <hr/> <p>7(B)</p> <p>The Figure 5A example shows items "within" and enclosed by container 302. However, <b>container 302 may "contain" items without those items actually being stored within the container. For example, the container 302 may reference items that are available elsewhere such as in other containers at remote sites.</b> Container 302 may reference items available at different times or only during limited times. Some items may be too large to store within container 302. Items may, for example, be delivered to the user in the form of a "live feed" of video at a certain time. Even then, the container 302 "contains" the live feed (by reference) in this example.</p> <p>'193 patent at 58:48-58.</p> <hr/> <p><u>Extrinsic Sources</u></p> <p>7(C)</p> <p><b>contain</b> <i>tr.v.</i> -tained, -tain-ing, -tains. 1. a. <b>To have within; hold.</b> b. To be capable of holding. 2. To have as component parts; include or comprise: <i>The album contains many memorable songs.</i> 3. a. To hold or keep within limits; restrain: <i>I could hardly contain my curiosity.</i> b. To halt the spread or development of; check: <i>Science sought an effective method of containing the disease.</i> 4. To check the expansion or influence of (a hostile power or ideology) by containment. 5. <i>Mathematics.</i> To be exactly divisible by. [Middle English <i>conteninen</i>, from Old French <i>contenir</i>, from Latin <i>continere</i> : <i>com-</i>, <i>com-</i> + <i>tenere</i>, to hold. See <i>ten-</i>.]--<b>con-tain'a-ble</b> <i>adj.</i></p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p><b>SYNONYM:</b> <i>contain, hold, accommodate.</i> These verbs mean to have within or have the capacity for having within. <i>Contain</i> means to have within or have as a part or constituent: <i>This drawer contains all the cutlery we own. The book contains some amusing passages. Polluted water contains contaminants.</i> <i>Hold</i> can be used in that sense but primarily stresses capacity for containing: <i>The pitcher holds two pints but contains only one.</i> <i>Accommodate</i> refers to capacity for holding comfortably: <i>The restaurant accommodates 50 customers. Four hundred inmates were crowded into a prison intended to accommodate 200.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 406.</p> |

|    | Claim Term /<br>Phrase   | InterTrust Evidence   |
|----|--|---|
| 8. | control (n.)<br><br>193.1, 193.11,<br>193.15, 193.19,<br>891.1 | <p><u>Patent Specifications</u></p> <p>8(A)</p> <p>Consumers 206, 208, 210 are each capable of receiving and using the programs created by video production studio 204—assuming, that is, that the video production studio or information utility 200 has arranged for these consumers to have appropriate “rules and controls” (control information) that give the consumers rights to use the programs.</p> <p>‘193 patent at 53:53-59.</p> <hr/> <p>8(B)</p> <p>The virtual distribution environment 100 prevents use of protected information except as permitted by the “rules and controls” (control information). For example, the “rules and controls” shown in Figure 2 may grant specific individuals or classes of content users 112 “permission” to use certain content. They may specify what kinds of content usage are permitted, and what kinds are not. They may specify how content usage is to be paid for and how much it costs. As another example, “rules and controls” may require content usage information to be reported back to the distributor 106 and/or content creator 102.</p> <p>‘193 patent at 56:26-36.</p> <hr/> <p>8(C)</p> <p>Objects may be classified in one sense based on whether the protection information is bound together with the protected information. For example, a container that is bound by its control(s) to a specific VDE node is called a “stationary object” (see Figure 18). A container that is not bound by its control information to a specific VDE node but rather carries sufficient control and permissions to permit its use, in whole or in part, at any of several sites is called a “Traveling Object”....</p> <p>‘193 patent at 129:52-60.</p> |

| Claim Term / Phrase   | InterTrust Evidence   |   |                  |                |                        |                        |
|---|---|---|------------------|----------------|------------------------|------------------------|
|   | <p>8(D)</p> <p>VDEF load modules, associated data, and methods form a body of information that for the purposes of the present invention are called <u>control information</u>. VDEF control information may be specifically associated with one or more pieces of electronic content and/or it may be employed as a general component of the operating system capabilities of a VDE installation.</p> <p>'193 patent at 18:36-42.</p> <hr/> <p>8(E)</p> <p>Failure information, including the elements listed below, may be saved along with details of the failure:</p> <table><tr><td><u>Control Information</u> Retained in an<br/>SPE on Access Failures</td></tr><tr><td><u>Object ID</u></td></tr><tr><td><u>User ID</u></td></tr><tr><td><u>Type of failure</u></td></tr><tr><td><u>Time of failure</u></td></tr></table> <p>This information may be analyzed to detect cracking attempts or to determine patterns of usage outside expected (and budgeted) norms. The audit trail histories in the SPU 500 may be retained until the audit is reported to the appropriate parties.</p> <p>'193 patent at 121:15-32.</p> <hr/> <p>8(F)</p> <p>In this embodiment, the additional memory may be provided by additional one or more integrated circuits that can be contained within a secure enclosure, such as a tamper resistant metal container or some form of a chip pack containing multiple integrated circuit components, and which impedes and/or evidences tampering attempts, and/or disables a portion or all of SPU 500 or associated <u>critical key and/or other control information</u> in the event of tampering.</p> <p>'193 patent at 169:5-13.</p> | <u>Control Information</u> Retained in an<br>SPE on Access Failures | <u>Object ID</u> | <u>User ID</u> | <u>Type of failure</u> | <u>Time of failure</u> |
| <u>Control Information</u> Retained in an<br>SPE on Access Failures |   |   |                  |                |                        |                        |
| <u>Object ID</u>  |   |   |                  |                |                        |                        |
| <u>User ID</u>  |   |   |                  |                |                        |                        |
| <u>Type of failure</u>  |   |   |                  |                |                        |                        |
| <u>Time of failure</u>  |   |   |                  |                |                        |                        |



|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="521 331 581 365"><b>8(G)</b></p> <p data-bbox="613 401 1321 470">... may involve preserving at least a portion of the control information (e.g., executable code such as load modules)</p> <p data-bbox="521 506 813 539">'193 patent at 33:12-14.</p> <hr/> <p data-bbox="521 621 581 655"><b>8(H)</b></p> <p data-bbox="613 690 1446 1276">VDE control information may, in part or in full, (a) represent control information directly put in place by VDE content control information pathway participants, and/or (b) comprise control information put in place by such a participant on behalf of a party who does not directly handle electronic content (or electronic appliance) permissions records information (for example control information inserted by a participant on behalf of a financial clearinghouse or government agency). Such control information methods (and/or load modules and/or mediating data and/or component assemblies) may also be put in place by either an electronic automated, or a semi-automated and human assisted, control information (control set) negotiating process that assesses whether the use of one or more pieces of submitted control information will be integrated into and/or replace existing control information (and/or chooses between alternative control information based upon interaction with in-place control information) and how such control information may be used.</p> <p data-bbox="521 1312 813 1346">'193 patent at 44:34-52.</p> <hr/> <p data-bbox="521 1428 570 1461"><b>8(I)</b></p> <p data-bbox="613 1497 1409 1667">In either embodiment, certain control information (software and parameter data) must be securely maintained within the SPU, and further control information can be stored externally and securely (e.g. in encrypted and tagged form) and loaded into said hardware SPU when needed.</p> <p data-bbox="521 1703 813 1736">'193 patent at 49:50-55.</p> <hr/> <p data-bbox="521 1818 570 1852"><b>8(J)</b></p> <p data-bbox="613 1887 1386 1917">Content control information governs content usage according to</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>criteria set by holders of rights to an object's contents and/or according to parties who otherwise have rights associated with distributing such content (such as governments, financial credit providers, and users).</p> <p>'193 patent at 15:46-50.</p> <hr/> <p><b>8(K)</b></p> <p>VDE's usage control information, for example, provide for property content and/or appliance related: usage authorization, usage auditing (which may include audit reduction), usage billing, usage payment, privacy filtering, reporting, and security related communication and encryption techniques.</p> <p>'193 patent at 15:33-38.</p> <hr/> <p><b>8(L)</b></p> <p>Control information delivered by, and/or otherwise available for use with, VDE content containers comprise (for commercial content distribution purposes) VDEF control capabilities (and any associated parameter data) for electronic content. These capabilities may constitute one or more "proposed" electronic agreements (and/or agreement functions available for selection and/or use with parameter data) that manage the use and/or the consequences of use of such content and which can enact the terms and conditions of agreements involving multiple parties and their various rights and obligations.</p> <p>'193 patent at 19:22-32.</p> <hr/> <p><b>8(M)</b></p> <p>... an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).</p> <p>'193 patent at 48:29-34.</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="526 338 586 373"><b>8(N)</b></p> <p data-bbox="615 407 1403 478">In the Figure 5A example, container 302 may also contain "rules and controls" in the form of:</p> <ul data-bbox="615 512 1003 617" style="list-style-type: none"> <li data-bbox="615 512 1003 548">(a) a "permissions record" 808;</li> <li data-bbox="615 548 1003 583">(b) "budgets" 308; and</li> <li data-bbox="615 583 1003 617">(c) "other methods" 1000.</li> </ul> <p data-bbox="615 653 1442 1066">Figure 5B gives some additional detail about permissions record 808, budgets 308 and other methods 1000. The "permissions record" 808 specifies the rights associated with the object 300 such as, for example, who can open the container 302, who can use the object's contents, who can distribute the object, and what other control mechanisms must be active. For example, permissions record 808 may specify a user's rights to use, distribute and/or administer the container 302 and its content. Permissions record 808 may also specify requirements to be applied by the budgets 308 and "other methods" 1000. Permissions record 808 may also contain security related information such as scrambling and descrambling "keys."</p> <p data-bbox="615 1102 1435 1310">"Budgets" 308 shown in Figure 5B are a special type of "method" 1000 that may specify, among other things, limitations on usage of information content 304, and how usage will be paid for. Budgets 308 can specify, for example, how much of the total information content 304 can be used and/or copied. The methods 310 may prevent use of more than the amount specified by a specific budget.</p> <p data-bbox="526 1346 802 1381">'193 patent at 59:1-25.</p> <hr/> <p data-bbox="526 1457 586 1493"><b>8(O)</b></p> <p data-bbox="615 1528 1435 1667">A distributed database may manage such a distributed repository resource environment and use VDE to secure the storing, communicating, auditing, and/or use of information through VDE's electronic enforcement of VDE controls.</p> <p data-bbox="526 1703 834 1738">'193 patent at 284:22-26.</p> <hr/> <p data-bbox="526 1814 586 1850"><b>8(P)</b></p> <p data-bbox="615 1885 1305 1921">ROS 602 provided by the preferred embodiment extends</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>conventional capabilities such as, for example, Access Control List (ACL) structures, to user and process defined events, including state transitions. ROS 602 may provide full control information over pre-defined and user-defined application events. These control mechanisms include "go/no-go" permissions, and also include optional event-specific executables that permit complete flexibility in the processing and/or controlling of events. This structure permits events to be individually controlled so that, for example, metering and budgeting may be provided using independent executables. <u>For example, ROS 602 extends ACL structures to control arbitrary granularity of information.</u> Traditional operating systems provide static "go-no go" control mechanisms at a file or resource level; ROS 602 extends the control concept in a general way from the largest to the smallest sub-element using a flexible control structure. ROS 602 can, for example, control the printing of a single paragraph out of a document file.</p> <p>'193 patent at 77:45-63.</p> <hr/> <p><b>8(Q)</b></p> <p><u>ROS 602 provided by the preferred embodiment permits secure modification and update of control information</u> governing each component. The control information may be provided in a template format such as method options to an end-user. An end-user may then customize the actual control information used within guidelines provided by a distributor or content creator.</p> <p>'193 patent at 77:64-78:3.</p> <hr/> <p><b>8(R)</b></p> <p>VDE <u>control information (e.g., methods)</u> that collectively control use of VDE managed properties (database, document, individual commercial product), <u>are either shipped with the content itself (for example, in a content container) and/or one or more portions of such control information is shipped to distributors and/or other users in separably deliverable "administrative objects."</u> A subset of the methods for a property may in part be delivered with each property while one or more other subsets of methods can be delivered</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="609 331 1393 401"><del>separately</del> to a user or otherwise made available for use (such as being available remotely by telecommunication means).</p> <p data-bbox="516 436 812 468">'193 patent at 43:26-37.</p> <hr/> <p data-bbox="516 552 574 583">8(S)</p> <p data-bbox="609 619 1429 758"><del>Many such load modules are inherently configurable, aggregatable, portable, and extensible</del> and singularly, or in combination (along with associated data), run as control methods under the VDE transaction operating environment.</p> <p data-bbox="516 793 812 825">'193 patent at 25:48-52.</p> <hr/> <p data-bbox="516 909 574 940">8(T)</p> <p data-bbox="609 976 1442 1528">Traveling objects can be used at a receiving VDE node electronic appliance 600 so long as either the appliance carries the correct budget or budget type (e.g. sufficient credit available from a clearinghouse such as a VISA budget) <del>either in general or for specific one or more users or user classes</del>, or so long as the traveling object itself carries with it sufficient budget allowance or an appropriate authorization (e.g., a stipulation that the traveling object may be used on certain one or more installations or installation classes or users or user classes where classes correspond to a specific subset of installations or users who are represented by a predefined class identifiers stored in a secure database 610). After receiving a traveling object, if the user (and/or installation) doesn't have the appropriate budget(s) and/or authorizations, then the user could be informed by the electronic appliance 600 (using information stored in the traveling object) as to which one or more parties the user could contact.</p> <p data-bbox="516 1564 823 1596">'193 patent at 131:33-50.</p> <hr/> <p data-bbox="516 1680 574 1711">8(U)</p> <p data-bbox="609 1747 1442 1917">[A]n object provider might allow users to redistribute copies of an object to their friends and associates (for example by physical delivery of storage media or by delivery over a computer network) such that if a friend or associate satisfies any certain criteria required for use of said object, he may do so.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p>For example, if a software program was distributed as a traveling object, a user of the program who wished to supply it or a usable copy of it to a friend would normally be free to do so. Traveling Objects have great potential commercial significance, since useful content could be primarily distributed by users and through bulletin boards, which would require little or no distribution overhead apart from registration with the "original" content provider and/or clearinghouse.</p> <p>The "out of channel" distribution may also allow the provider to receive payment for usage and/or otherwise maintain at least a degree of control over the redistributed object. Such certain criteria might involve, for example, the registered presence at a user's VDE node of an authorized third party financial relationship, such as a credit card, along with sufficient available credit for said usage.</p> <p>Thus, if the user had a VDE node, the user might be able to use the traveling object if he had an appropriate, available budget available on his VDE node (and if necessary, allocated to him), and/or if he or his VDE node belonged to a specially authorized group of users or installations and/or if the traveling object carried its own budget(s).</p> <p>'193 patent at 131:59-132:18.</p> <hr/> <p><b>8(V)</b></p> <p>VDE supports multiple differing hierarchies of client organization control information wherein an organization client administrator distributes control information specifying the usage rights of departments, users, and/or projects. Likewise, a department (division) network manager can function as a distributor (budgets, access rights, etc.) for department networks, projects, and/or users, etc.</p> <p>'193 patent at 33:63-34:3.</p> <hr/> <p><b><u>File Histories</u></b></p> <p><b>8(W)</b></p> <p>Claims . . . are rejected under 35 U.S.C. 102(b) as being anticipated by Lofberg (4,595,950).</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>The recited first device and its operation matches that of the rent terminal. . . . The information including at least one control is the personal identification information, see col. 3, lines 60-68 and col. 4, lines 64-68 and col. 13, lines 1-11. . . . The second device is the user station. The rent terminal determines whether the digital file may be copied and stored on the second device, see col. 9, lines 1-8 and col. 12, lines 43-49. The second device renders the digital file through its output only upon the data carrier having the information recorded therein and governing the use of the digital file is transferred to the second device.</p> <p>'193 Patent File History, 6/7/00 Office Action, p. 2.</p> <hr/> <p><b>8(X)</b></p> <p>Claims . . . are rejected . . . as being anticipated by Karp (4,866,769).</p> <p>. . . The first device is a personal computer that is allowed access to the software by virtue of an encoded checkword derived from a source ID on the diskette and the personal computer ID, see Abstract. The information including at least one control is the list of checkwords stored in association with the digital file, see col. 5, line 60 through col. 6, line 11. A second device is represented by a second checkword stored in the list, see col. 8, lines 1-18. The determination of whether the digital file may be copied and stored by a second device is dependent on whether a checkword for the second device is allowed.</p> <p>'193 Patent File History, 6/7/00 Office Action, pp. 3-4.</p> <hr/> <p><b>8(Y)</b></p> <p>Claims 58-59 are rejected . . . as being anticipated by Schull [5,509,070].</p> <p>The Schull reference describes a system for distribution, registration and purchase of software. . . . The identified control is the need for a valid password to unlock the advanced features of the copied</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="618 338 1427 405">software. Column 7, line 10 through column 8, line 9 describe the generation and assignment of the target IDs and passwords.</p> <p data-bbox="526 443 1218 474">USP 5,915,019 File History, 7/28/97 Office Action, p. 3.</p> <hr data-bbox="521 514 1458 520"/> <p data-bbox="526 558 581 590"><b>8(Z)</b></p> <p data-bbox="618 627 1443 831">[Okano, 5,504,818] describes a system using cryptography for processing various digital objects. Figure 3 and column 6, line 33 disclose where a protected object may have embedded additional elements (security code attributes) to associate a control on the object. The control would restrict information according to security levels.</p> <p data-bbox="526 869 1214 900">USP 5,915,019 File History, 7/28/97 Office Action, p. 3.</p> <hr data-bbox="521 940 1458 947"/> <p data-bbox="526 984 602 1016"><b>8(AA)</b></p> <p data-bbox="618 1054 1443 1291">A comparison of independent claim 7 to Fischer to derive the similarities and differences between the claimed invention and the prior art follows. Figure 1 of Fischer shows various terminals connected via a communications channel 12. Terminal A, as a first apparatus recited in claim 7, includes user controls as per keyboard / crt. 4; communications port, see modem and communications channel 12; a processor as processor with main memory, 2....</p> <p data-bbox="526 1329 1097 1360">'683 File History, 11/12/99 Office Action, p. 4.</p> |



|    | Claim Term / Phrase                            | InterTrust Evidence  |
|----|--|--|
| 9. | controlling, control (v.)<br><br>193.1, 861.58 | <p><u>Patent Specifications</u></p> <p>9(A)</p> <p>Secondary storage 652 in this example stores code and data used by CPU 654 and/or SPU 500 to <u>control</u> the overall operation of electronic appliance 600.</p> <p>'193 patent at 62:58-60.</p> <hr/> <p>9(B)</p> <p>The other CPU(s) 654 may be any centrally <u>controlling</u> logic arrangement, such as for example, a microprocessor, other microcontroller, and/or array or other parallel processor.</p> <p>'193 patent at 64:55-58.</p> <hr/> <p>9(C)</p> <p>A shared address/data bus arrangement 536 may transfer information between these various components under <u>control</u> of microprocessor 520 and/or DMA controller 526.</p> <p>'193 patent at 65:35-38.</p> <hr/> <p>9(D)</p> <p>In some implementations, a separate arithmetic accelerator 544 may be omitted and any necessary calculations may be performed by microprocessor 520 under software <u>control</u>.</p> <p>'193 patent at 68:46-49.</p> <hr/> <p>9(E)</p> <p>DMA controller 526 <u>controls</u> information transfers over address/data bus 536 without requiring microprocessor 520 to process each individual data transfer.</p> <p>'193 patent at 68:51-53.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="527 331 586 365">9(F)</p> <p data-bbox="618 401 1419 470">In the preferred embodiment, to <u>control</u> access to clearinghouses, users are assigned account numbers at clearinghouses.</p> <p data-bbox="527 506 837 539">'93 patent at 268:29-31.</p> <hr data-bbox="527 577 1463 583"/> <p data-bbox="527 619 586 653">9(G)</p> <p data-bbox="618 709 1451 1255">... plural, different control models regulating the use and/or auditing of either the same specific copy of electronic information content and/or differently regulating different copies (occurrences) of the same electronic information content. Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments. This includes supporting variable control information for budgeting and auditing usage as applied to a variety of predefined increments of electronic information, including <u>employing a variety of different budgets and/or metering increments for a given electronic information deliverable</u> for: billing units of measure, credit limit, security budget limit and security content metering increments, and/or market surveying and customer profiling content metering increments.</p> <p data-bbox="527 1291 821 1325">'93 patent at 28:19-37.</p> <hr data-bbox="527 1362 1463 1369"/> <p data-bbox="527 1407 586 1440">9(H)</p> <p data-bbox="618 1476 1458 1923">... support the flowing of content control information through different "branches" of content control information handling so as to accommodate, under the present invention's preferred embodiment, diverse controlled distributions of VDE controlled content. <u>This allows different parties to employ the same initial electronic content with differing (perhaps competitive) control strategies.</u> In this instance, a party who first placed control information on content can make certain control assumptions and these assumptions would evolve into more specific and/or extensive control assumptions. These control assumptions can evolve during the branching sequence upon content model participants submitting control information changes, for example, for use in "negotiating" with "in place" content control information. This can result in new or</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>modified content control information and/or it might involve the selection of certain one or more already "in-place" content usage control methods over in-place alternative methods, as well as the submission of relevant control information parameter data. This form of evolution of <u>different control information sets applied to different copies of the same electronic property content</u> and/or appliance results from VDE control information flowing "down" through different branches in an overall pathway of handling and control and being modified differently as it diverges down these different pathway branches.</p> <p>'193 patent at 31:29-56.</p> <hr/> <p>9(I)</p> <p><u>... multiple simultaneous control models for the same content property and/or property portion.</u> This allows, for example, for concurrent business activities which are dependent on electronic commercial product content distribution, such as acquiring detailed market survey information and/or supporting advertising, both of which can increase revenue and result in lower content costs to users and greater value to content providers. <u>Such control information and/or overall control models may be applied, as determined or allowed by control information, in differing manners to different participants</u> in a pathway of content, reporting, payment, and/or related control information handling. <u>VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities, and/or to different parties in a content and/or appliance usage model, such that different parties (or classes of VDE users, for example) are subject to differing control information managing their use of electronic information content.</u> For example, <u>differing control models based on the category of a user as a distributor of a VDE controlled content object or an end-user of such content may result in different budgets being applied.</u> Alternatively, for example, a one distributor may have the right to distribute a different array of properties than another distributor (from a common content collection provided, for example, on optical disc). <u>An individual, and/or a class or other grouping of end-users, may have different costs (for example, a student, senior citizen, and/or poor citizen user of content who may be provided with the same or differing discounts) than a "typical" content user.</u></p> <p>'193 patent at 30:42-31:7.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="526 338 581 373">9(J)</p> <p data-bbox="613 407 1451 1037">Such different application of control information may also result from content control information specifying that a certain party or group of parties shall be subject to content control information that differs from another party or group of parties. For example, content control information for a given piece of content may be stipulated as senior information and therefore not changeable, might be put in place by a content creator and might stipulate that national distributors of a given piece of their content may be permitted to make 100,000 copies per calendar quarter, so long as such copies are provided to bona fide end-users, but may pass only a single copy of such content to a local retailers and the control information limits such a retailer to making no more than 1,000 copies per month for retail sales to end-users. In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).</p> <p data-bbox="519 1071 812 1106">'193 patent at 48:15-35.</p> <hr/> <p data-bbox="516 1182 581 1218">9(K)</p> <p data-bbox="607 1251 1442 1946">In this example, as illustrated in Figure 80, user B may receive control information associated with creator A's content from distributor A and/or user/distributor B. In either case, user B may be able to establish their own control information on DA(CA) and/or UDB(UDA(DA(CA))), respectively (if allowed by such control information. The resulting set(s) of control information UB(DA(CA)) and/or UB(UDB(UDA(DA(CA)))) respectively may represent different control scenarios, each of which may have benefits for user B. As described in connection with an earlier example, user B may have received control information from user/distributor B along a chain of handling including user/distributor A that bases fees on the number of minutes that user B makes use of creator A's content (and requiring user/distributor A to pay fees of \$15 per month per user to distributor A regardless of the amount of usage by user B in a calendar month). This may be more favorable under some circumstances than the fees required by a direct use of control information provided by distributor A, but may also have the disadvantage of an exhausted chain of redistribution and, for example, further usage information reporting requirements included in UDB(UDA(DA(CA))). If the two sets of</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>control information DA(CA) and UDB(UDA(DA(CA))) permit (e.g. do not require exclusivity enforced, for example, by using a registration interval in an object registry used by a secure subsystem of user B's VDE installation to prevent deregistration and reregistration of different sets of control information related to a certain container (or registration of plural copies of the same content having different control information and/or being supplied by different content providers) within a particular interval of time as an aspect of an extended agreement for a chain of handling and control reflected in DA(CA) and/or UDB(UDA(DA(CA))) , user B may have both sets of control information registered and may make use of the set that they find preferable under a given usage scenario.</p> <p>'193 patent at 306:30-65.</p> <hr/> <p><b>9(L)</b></p> <p>For example, user/distributor A may receive control information CB that includes a requirement that user/distributor A pay creator B for content decrypted by user/distributor A (and any participant receiving distributed and/or redistributed control information from user/distributor A) at the rate of \$0.50 per kilobyte. As indicated above, user/distributor A also may receive control information associated with creator B's VDE content container from distributor A. In this example, user/distributor A may have a choice between paying a "rental" fee through a chain of handling passing through distributor A, and a fee based on the quantity of decryption through a chain of handling direct to creator B. In this case, user/distributor A may have the ability to choose to use either or both of CB and DA(CB).</p> <p>'193 patent at 308:29-42.</p> <hr/> <p><b>9(M)</b></p> <p>As illustrated in Figure 81, in this example, user B may receive control information associated with creator B's VDE content container from six different sources: CB directly from creator B, DA(CB) from distributor A, UDB(UDA(DA(CB))) and/or UDB(UDA(CB)) from user/distributor B, DC(CB) from distributor C, and/or DB(DC(CB)) from distributor B. This represents six chains of handling through which user B may enter into extended agreements with other participants in this example. Two of these</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>chains pass through user/distributor B. Based on a VDE negotiation between user/distributor B and user B, an extended agreement may be reached (if permitted by control information governing both parties) that reflects the conditions under which user B may use one or both sets of control information. In this example, two chains of handling and control may "converge" at user/distributor B, and then pass to user B (and if control information permits, later diverge once again based on distribution and/or redistribution by user B).</p> <p>'193 patent at 308:48-65.</p> <hr/> <p>9(N)</p> <p>User B may, in this example, receive a VDE content container from distributor C that is comprised of VDE objects created by creator B, creator C, and creator D. In addition, user B may receive a VDE content container from distributor B that contains the same content created by creator B, creator C, and creator D in addition to one or more extracted/embedded portions of content created by creator E. User B may base decisions concerning which of such containers they choose to use (including which embedded containers she may wish to use), and under which circumstances, based on, for example, the character of such extracted/embedded portions (e.g. multimedia presentations illustrating potential areas of interest in the remainder of the content, commentary explaining and/or expositing other elements of content, related works, improved application software delivered as an element of content, etc.); the quality, utility, and/or price (or other attributes of control information) of such portions; and other considerations which distinguish the containers and/or content control information received, in this example, from distributor B and distributor C.</p> <p>'193 patent at 312:11-31.</p> <hr/> <p><u>Extrinsic Sources</u></p> <p>9(O)</p> <p><u>control</u> <i>tr. v. -trolled, -trol-ling, -trols.</i> 1. <u>To exercise authoritative or dominating influence over, direct.</u> See Synonyms at <u>conduct</u>. 2. To hold in restraint; check: <i>struggled to control my temper; regulations intended to control prices.</i> 3. a. To verify or regulate (a scientific experiment) by conducting a parallel</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>experiment or by comparing with another standard. <b>b.</b> To verify (an account, for example) by using a duplicate register for comparison. <b>—control</b> <i>n.</i> <b>1.</b> Authority or ability to manage or direct: <i>lost control of the skidding car; the leaders in control of the country.</i> <b>2.</b> <i>Abbr. cont., contr.</i> <b>a.</b> One that controls; a controlling agent, device, or organization. <b>b.</b> Often <b>controls.</b> An instrument or set of instruments used to operate, regulate, or guide a machine or vehicle. <b>3.</b> A restraining device, measure, or limit; a curb: <i>a control on prices; price controls.</i> <b>4.</b> <b>a.</b> A standard of comparison for checking or verifying the results of an experiment. <b>b.</b> An individual or group used as a standard of comparison in a control experiment. <b>5.</b> An intelligence agent who supervises or instructs another agent. <b>6.</b> A spirit presumed to speak or act through a medium. [Middle English <i>controllen</i>, from Anglo-Norman <i>contreroller</i>, from Medieval Latin <i>contrarotulare</i>, to check by duplicate register, from <i>contrarotulus</i>, duplicate register : Latin <i>contra-</i>, <i>contra-</i> + Latin <i>rotulus</i>, roll, diminutive of <i>rota</i>, wheel. See <i>ret-</i>.] <b>—con-trol'la-bil'i-ty</b> <i>n.</i> <b>—con-trol'la-ble</b> <i>adj.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 410.</p> |

|     | Claim Term / Phrase  | InterTrust Evidence  |
|-----|--|--|
| 10. | copy, copied, copying<br><br>193.1, 193.11, 193.15, 193.19 | <p><u>Patent Specifications</u></p> <p>10(A)</p> <p>In some circumstances, a VDE administrator may require that a <u>copy (partial or complete)</u> of the back up files be transmitted to it within an administrative object to check for indications of fraudulent activities by the user.</p> <p>'193 patent at 167:63-67.</p> <hr/> <p>10(B)</p> <p>When a user needs to access a particular VDE object 300, her electronic appliance 600 could issue a request over network 672 <u>to obtain a copy of the object</u>. The "VDE server" could deliver all or a <u>portion of the requested object 300</u> in response to the request.</p> <p>'193 patent at 226:11-16.</p> <hr/> <p>10(C)</p> <p>Expiration dates cannot be used effectively to prevent substitution of the <u>previous copy</u> of a budget UDE 1200. To secure these frequently updated items, a transaction tag is generated and included in the encrypted item each time that item is updated.</p> <p>'193 patent at 143:14-18.</p> <hr/> <p>10(D)</p> <p>For example, author 3306A may have required that the repository <u>encrypt each copy of shipped content using a different encryption key or keys</u> in order to help maintain greater protection for content (e.g. in case an encryption key was "cracked" or inadvertently disclosed, the "damage" could be limited to the portion(s) of that specific copy of a certain content deliverable).</p> <p>'193 patent at 288:46-52.</p> <hr/> |



|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="527 344 602 375"><b>10(E)</b></p> <p data-bbox="618 415 1425 485">electronic testing will allow users to receive a <u>copy (encrypted or unencrypted)</u> of their test results when they leave the test sessions.</p> <p data-bbox="527 520 834 552">'93 patent at 319:13-15.</p> <hr/> <p data-bbox="527 632 602 663"><b>10(F)</b></p> <p data-bbox="618 703 1455 909">transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output, the portion of said digital file transferred to said second device representing a version of said digital file which, when rendered at said second device, provides a level of quality lower than the level of quality provided when said digital file is rendered at said first device;</p> <p data-bbox="527 945 870 976">'93 patent at 323:64-324:4.</p> <hr/> <p data-bbox="527 1056 602 1087"><b>10(G)</b></p> <p data-bbox="618 1127 1442 1297">For example, if the audit information received by the clearinghouse is legitimate, then the clearinghouse may send an administrative object to the end user's electronic appliance 600 <u>requesting the electronic appliance to delete and/or compress the audit information that has been transferred.</u></p> <p data-bbox="527 1333 829 1365">'93 patent at 162:10-15.</p> <hr/> <p data-bbox="527 1444 602 1476"><b>10(H)</b></p> <p data-bbox="618 1516 1446 1829">[A] user (the "originating user") may wish to place an "originator controlled" ("ORCON") restriction on a certain document, such that the document may be transmitted and used only by those specific other users whom he designates (and only in certain, expressly authorized ways). Such a restriction may be flexible if the "distribution list" could be modified after the creation of the document, specifically in the event of <u>someone requesting permission from the originating user to transmit the document outside the original list of authorized recipients.</u></p> <p data-bbox="527 1864 829 1896">'93 patent at 278:11-21.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p><b>10(I)</b></p> <p>Commercial content repository 200g may store content securely such that users may obtain such, when any required conditions are met, content from the repository 200g. The distribution permissions 3502 may, for example, permit commercial content repository 200g to create redistribution permissions and/or usage permissions 3500, 3502 using a VDE protected subsystem within certain restrictions described in content control information received from creator 102 (e.g., not to exceed a certain number of copies, requiring certain payments by commercial content repository 200g to creator 102, requiring recipients of such permissions to meet certain reporting requirements related to content usage information, etc.). Such content control information may be stored at the repository installation and be applied to unencrypted content as it is transmitted from said repository in response to a user request, wherein said content is placed into a VDE container as a step in a secure process of communicating such content to a user.</p> <p>'193 patent at 316:16-37.</p> <hr/> <p><b>10(J)</b></p> <p>37. A method as in claim 36, further comprising:</p> <p>at some point after said transferring step, taking at least one action to render said copy of said first digital file unuseable at said second device; and</p> <p>at said first digital device, removing said encumbrance on said budget,</p> <p>said removal including increasing the number of copies of said first digital file authorized by said budget.</p> <p>'193 patent at 325:32-40.</p> <hr/> <p><b><u>Extrinsic Sources</u></b></p> <p><b>10(K)</b></p> <p>copy to reproduce data in a new location or other destination, leaving the source data unchanged, although the physical form of</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>the result may differ from that of the source; for example, to make a duplicate of all the programs or data on a disk, or to copy a graphic screen image to a printer.</p> <p>Spencer, Personal Computer Dictionary (Camelot Publishing, 1995), p. 47.</p> <hr/> <p><b>10(L)</b></p> <p><b>copy</b> 1. The material, including text, graphic images, pictures, and artwork, to be assembled for printing. <del>To reproduce part of a document at another location in the document or in another document.</del></p> <p>Webster's New World Dictionary of Computer Terms, 6th ed. (1997), p. 118.</p> <hr/> <p><b>10(M)</b></p> <p><b>copy</b> <i>n., pl. -ies.</i> 1. An imitation or <del>reproduction</del> of an original; a <del>duplicate</del>: <i>a copy of a painting; made two copies of the letter.</i> 2. One specimen or example of a printed text or picture: <i>an autographed copy of a novel.</i> 3. <i>Abbr. c., C.</i> Material, such as a manuscript, that is to be set in type. 4. The words to be printed or spoken in an advertisement. 5. Suitable source material for journalism: <i>Celebrities make good copy.</i> <b>-copy</b> <i>v. -ied, -ying, -ies</i> <b>-tr.</b> 1. To make a reproduction or copy of. 2. To follow as a model or pattern; imitate. See Synonyms at <b>imitate</b>. <b>-intr.</b> 1. To make a copy or copies. 2. To admit of being copied: <i>colored ink that does not copy well.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 416.</p> |

|     | Claim Term / Phrase   | InterTrust Evidence   |
|-----|-----------------------|---|
| 11. | derive<br><br>900.155 | <p><u>Patent Specifications</u></p> <p><b>11(A)</b></p> <p>Whenever CPU/SPU 2650 enters or leaves the "SPU" mode, the transition is performed in such a way that no information contained in the secure memory 532, 534 or <u>derived</u> from it (e.g., stored in registers or a cache memory associated with microprocessor 2652) while in the "SPU" mode can be exposed by microprocessor 2652 operations that occur in the "normal" mode.</p> <p>'900 patent at 75:30-36.</p> <hr/> <p><b>11(B)</b></p> <p>In some example implementations, interrupts may be enabled while CPU/SPU 2650 is operating in the "SPU" mode similarly interrupts and returns from interrupts while in the "SPU" mode may allow transitions from "SPU" mode to "normal" mode and back to "SPU" mode without exposing the content of secure memory 532, 534 or the content of registers or other memory associated with microprocessor 2652 that may contain information <u>derived</u> from secure mode operation.</p> <p>'900 patent at 75:41-49.</p> <hr/> <p><b>11(C)</b></p> <p>For example, during PPE 650 operation, the internal state of the PPE is constantly being updated. During each interaction with a trusted server, PPE 650 (and the trusted server) may test the internal state of PPE 650 to determine whether it could be <u>derived</u> from the internal state last seen by the trusted server for this particular PPE 650 instance. If it could not, the result may be taken as indicating a replay attack of some sort, and an appropriate action can be taken (see Figure 69L, block 3592, 3594, 3596).</p> <p>'900 patent at 247:4-12.</p> <hr/> <p><b>11(D)</b></p> <p>For example, the counter could be repeated hashing (e.g., with</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p>MD5) of a value that is stored redundantly in several different locations within the operational materials 3472 and secure database 610 - so that the trusted server could verify that the current value can be <u>derived</u> (e.g., by repeated MD5 applications) from a previous value.</p> <p>'900 patent at 247:20-26.</p> <hr/> <p><b><u>Extrinsic Sources</u></b></p> <p><b>11(E)</b></p> <p><b>derive:</b> v. <b>de-rived, de-riv-ing, de-rives.</b> v. <i>tr.</i> <b>1.</b> <u>To obtain or receive from a source.</u> <b>2.</b> <u>To arrive at by reasoning; deduce or infer: derive a conclusion from facts.</u> <b>3.</b> To trace the origin or development of (a word). <b>4. Chemistry.</b> To produce or obtain (a compound) from another substance by chemical reaction.v. <i>intr.</i> To issue from a source; originate. See Synonyms at <b>stem</b><sup>1</sup>. [Middle English <i>deriven</i>, to be derived from, from Old French <i>deriver</i>, from Latin <i>derivare</i>, to derive, draw off : <i>de-</i>, <i>de-</i> + <i>rivus</i>, stream. See <i>rei-</i>.]--<b>de-riv'a-ble</b> <i>adj.</i> --<b>de-riv'er</b> <i>n.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 504.</p> |

|     | Claim Term / Phrase      | InterTrust Evidence   |
|-----|--------------------------|---|
| 12. | designating<br><br>721.1 | <p><u>Patent Specifications</u></p> <p>12(A)</p> <p>Figures 11A-11C show how a verifying authority can use different digital signatures to designate the same or different load modules as being appropriate for execution by different assurance level electronic appliances;</p> <p>'721 patent at 7:66-8:2.</p> <hr/> <p>12(B)</p> <p>In one of its roles or instances, object submittal manager 774 provides a user interface 774a that allows the user to create an object configuration file 1240 specifying certain characteristics of a VDE object 300 to be created. This user interface 774a may, for example, allow the user to specify that she wants to create an object, allow the user to designate the content the object will contain, and allow the user to specify certain other aspects of the information to be contained within the object (e.g., rules and control information, identifying information, etc.).</p> <p>'193 patent at 103:11-20.</p> <hr/> <p>12(C)</p> <p>Control sets 914 exist in two types in VDE 100: common required control sets which are given designations "control set 0" or "control set for right," and a set of control set options.</p> <p>'193 patent at 150:30-33.</p> <hr/> <p>12(D)</p> <p>The classification attributes may designate the overall level of sensitivity of the document as an element of an ordered set. For example, the set "unclassified," "confidential," "secret," "top secret" might be appropriate in a government setting, and the set "public," "internal," "confidential," "registered confidential" might be appropriate in a corporate setting.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>The compartment attributes may <u>designate</u> the document's association with one or more specific activities within the organization, such as departmental subdivisions (e.g., "research," "development," "marketing") or specific projects within the organization.</p> <p>Each person using an electronic appliance 600 would be assigned, by an authorized user, a set of permitted sensitivity attributes to <u>designate</u> those documents, or one or more portions of certain document types, which could be processed in certain one or more ways, by the person's electronic appliance. A document's sensitivity attribute would have to belong to the user's set of permitted sensitivity values to be accessible.</p> <p>In addition, the organization may desire to permit users to exercise control over specific documents for which the user has some defined responsibility. As an example, a user (the "originating user") may wish to place an "originator controlled" ("ORCON") restriction on a certain document, such that the document may be transmitted and used only by those specific other users whom he <u>designates</u> (and only in certain, expressly authorized ways).</p> <p>'193 patent at 277:56-278:16.</p> <hr/> <p><b>12(E)</b></p> <p>A document may have an attribute <u>designating</u> its originator and requiring an explicit permission to be granted by an originator before the document's content could be viewed.</p> <p>'193 patent at 280:1-4.</p> <hr/> <p><b><u>Extrinsic Sources</u></b></p> <p><b>12(F)</b></p> <p><b><u>designate</u> tr. v. -nated, -nating, -nates. (1) <u>To indicate or specify; point out.</u> (2) <u>To give a name or title to; characterize.</u> (3) To select and set aside for a duty, an office, or a purpose. See Synonyms at <b>allocate, appoint.</b></b></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 506.</p> |

|     | Claim Term /<br>Phrase    | InterTrust Evidence   |
|-----|---------------------------|---|
| 13. | device class<br><br>721.1 | <p><u>File Histories</u></p> <p>13(A)</p> <p>... Applicants respectfully submit that some of the terms cited by the Examiner as "indefinite" are either well-known by persons skilled in the art or inherently clear. For example ... the term "class" is used as part of the phrase "device class." Applicants respectfully submit that "device class" is inherently clear, meaning a group of devices which share at least one attribute.</p> <p>'721 Patent File History, 4/13/99 Response, p. 14.</p> |



|     | Claim Term / Phrase   | InterTrust Evidence  |
|-----|---|--|
| 14. | <p><b>digital signature, digitally signing</b></p> <p>721.1</p> | <p><u><b>Patent Specifications</b></u></p> <p><b>14(A)</b></p> <p>A verifying authority <u>digitally signs</u> and “certifies” those load modules or other executables it has verified <u>(using a public key based digital signature and/or certificate based thereon, for example).</u></p> <p>Protected execution spaces such as protected processing environments can be programmed or otherwise conditioned to accept only those load modules or other executables bearing a <u>digital signature/certificate of an accredited (or particular) verifying authority.</u></p> <p>‘721 patent at 4:64-5:5.</p> <hr/> <p><b>14(B)</b></p> <p>In accordance with another aspect provided by the present invention, an execution environment protects itself by deciding — based on digital signatures, for example — which load modules or other executables it is willing to execute. <u>A digital signature allows the execution environment to test both the authenticity and the integrity of the load module or other executables, as well permitting a user of such executables to determine their correctness with respect to their associated specifications or other description of their behavior, if such descriptions are included in the verification process.</u></p> <p>‘721 patent at 6:5-15.</p> <hr/> <p><b>14(C)</b></p> <p>A verifying authority may digitally sign load modules or other executables with a digital signature that indicates or implies assurance level. <u>A verifying authority can use digital signature techniques to distinguish between assurance levels. As one example, each different digital signature may be encrypted using a different verification key and/or fundamentally different encryption, one-way hash and/or other techniques.</u> A protected processing environment or other secure execution space protects itself by</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>executing only those load modules or other executables that have been digitally signed for its corresponding assurance level.</p> <p>'721 patent at 6:42-52.</p> <hr/> <p><b>14(D)</b></p> <p>Figure 5 shows how a verifying authority can create a certifying digital signature</p> <p>Figure 6 shows how a protected processing environment can securely authenticate a verifying authority's digital signature to guarantee the integrity of the corresponding load module;</p> <p>Figure 7 shows how several different digital signatures can be applied to the same load module;</p> <p>Figure 8 shows how a load module can be distributed with multiple digital signatures</p> <p>'721 patent at 7:47-57.</p> <hr/> <p><b>14(E)</b></p> <p>The two digital signature algorithms in widespread use today (RSA and DSA) are based on distinct mathematical problems (factoring in the case of RSA, discrete logs for DSA).</p> <p>'721 patent at 15:31-34.</p> <hr/> <p><b>14(F)</b></p> <p>There exist many well known processes for creating digital signatures. One example is the Digital Signature Algorithm (DSA). DSA uses a public-key signature scheme that performs a pair of transformations to generate and verify a digital value called a signature.</p> <p>'721 patent at 10:60-64.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="527 338 760 369"><u>Extrinsic Sources</u></p> <p data-bbox="527 407 605 438">14(G)</p> <p data-bbox="618 476 1446 716"><b>digital signature</b> In data security, a data block appended to a message, or a complete encrypted message, such that the recipient can authenticate the message contents and/or prove that it could only have originated with the purported sender. The digital signature is a function of: (a) the message, transaction or document to be signed; (b) secret information known only to the sender; and (c) public information employed in the validation process.</p> <p data-bbox="618 722 1446 1062">Message authentication enables the receiver of a message to ensure that the contents cannot be changed accidentally or deliberately by a third party. However, since both the sender and the receiver share the same secret information there is no method of resolving disputes. The receiver can compute the authenticator and could therefore change a message, or forge a new message, develop the authenticator and claim that it was transmitted by the sender sharing the same secret key for authentication. Conversely the sender could disown an authenticated message and claim that the receiver produced a forged message using the common secret key.</p> <p data-bbox="618 1068 1446 1478">The essence of a digital signature is that the receiver must be able to prove that a message originated with a given sender, but must not be able to construct the signed message. Thus the sender requires secret information to construct the signed message and the receiver must be able to access public information for use in the validation of the message. In the case of a dispute the receiver must be in a position to supply non-secret information to a judge (i.e., the signed message and the publicly available information) in order to prove the authentication and origin of the message. <i>Compare</i> DYNAMIC PASSWORD. <i>See</i> MESSAGE AUTHENTICATION, PUBLIC KEY CRYPTOGRAPHY, RSA. <i>Synonymous with</i> ELECTRONIC SIGNATURE.</p> <p data-bbox="527 1518 1398 1587">Dictionary of Information Technology, 3d ed. (Van Nostrand Reinhold, 1989), pp. 160-161.</p> <hr data-bbox="521 1623 1456 1629"/> <p data-bbox="521 1667 1386 1698"><u>Citations from Sources Designated by Microsoft under PLR 4-2(b)</u></p> <p data-bbox="521 1736 599 1768">14(H)</p> <p data-bbox="612 1806 1446 1875"><b>Digital signature</b> A string of characters that can be generated only by an agent that knows some secret, and hence provides evidence</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>that such an agent must have generated it.</p> <p>Neumann, Computer Related Risks (ACM Press, 1995), p. 345.</p> <hr/> <p><b>14(I)</b></p> <p>Another way to check your files for unauthorized tampering is to derive a signature for each file, and to compare that signature against a known value. A file signature is a function of the contents and properties of the file. A signature is relatively easy to calculate, but difficult to forge.</p> <p>Garfinkel et al., Practical Unix Security (O'Reilly &amp; Associates, 1991), p. 122.</p> |

|     | Claim Term / Phrase   | InterTrust Evidence   |
|-----|---|---|
| 15. | <p data-bbox="277 331 464 443"><b>executable, executable programming</b></p> <p data-bbox="277 489 456 552">721.34, 912.8, 912.35</p> | <p data-bbox="526 331 805 363"><u><b>Patent Specifications</b></u></p> <p data-bbox="526 405 602 436"><b>15(A)</b></p> <p data-bbox="615 474 1455 957"> The next section of load module 1100 is an encrypted executable body 1106 that contains one or more blocks of encrypted code. Load modules 1100 are preferably coded in the "native" instruction set of their execution environment for efficiency and compactness. SPU 500 and platform providers may provide versions of the standard load modules 1100 in order to make their products cooperate with the content in distribution mechanisms contemplated by VDE 100. The preferred embodiment creates and uses native mode load modules 1100 in lieu of an interpreted or "p-code" solution to optimize the performance of a limited resource SPU. However, when sufficient SPE (or HPE) resources exist and/or platforms have sufficient resources, these other implementation approaches may improve the cross platform utility of load module code. </p> <p data-bbox="526 999 833 1031">'193 patent at 141:42-56.</p> <hr/> <p data-bbox="526 1108 597 1140"><b>15(B)</b></p> <p data-bbox="615 1178 1455 1346"> The load module or other executable is preferably constructed using a programming language (e.g., languages such as Java and Python) and/or design/implementation methodology (e.g., Gypsy, FDM) that can facilitate automated analysis, validation, verification, inspection, and/or testing. </p> <p data-bbox="526 1388 800 1419">'721 patent at 5:34-39.</p> <hr/> <p data-bbox="526 1497 751 1528"><u><b>Extrinsic Sources</b></u></p> <p data-bbox="526 1570 597 1602"><b>15(C)</b></p> <p data-bbox="615 1640 1422 1734"> <b>executable</b> <i>adj.</i> Of, pertaining to, or being a program file that can be run. Executable files have extensions such as .bat, .com, and .exe. </p> <p data-bbox="615 1776 1390 1839"> <b>executable</b> <i>n.</i> A program file that can be run, such as file0.bat, file1.exe, or file2.com. </p> <p data-bbox="615 1881 1365 1913"> <b>executable program</b> <i>n.</i> A program that can be run. The term </p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>usually applies to a compiled program translated into machine code in a format that can be loaded into memory and run by a computer's processor. In interpreter languages, an executable program can be source code in the proper format. <i>See also</i> code (definition 1), compiler (definition 2), computer program, interpreter, source code.</p> <p>Microsoft Computer Dictionary, 3d ed. (Microsoft Press, 1997), p. 182.</p> |

|     | Claim Term / Phrase                        | InterTrust Evidence   |
|-----|--|---|
| 16. | host processing environment<br><br>900.155 | <p><u>Patent Specifications</u></p> <p><b>16(A)</b></p> <p>Personal computer 4116 in this example is also provided with a secure processing unit 500 or <del>software based HPE 655</del> (See Figure 12) to provide secure, tamper-resistant trusted processing.</p> <p>'683 patent at 20:16-19.</p> <hr/> <p><b>16(B)</b></p> <p><del>"Protected Processing Environment" ("PPE") 650 may refer generally to SPE 503 and/or HPE 655.</del> Hereinafter, unless context indicates otherwise, references to any of "PPE 650," "HPE 655" and "SPE 503" may refer to each of them.</p> <p>'193 patent at 105:18-22; '900 patent at 112:48-52.</p> <hr/> <p><b>16(C)</b></p> <p>As discussed above in connection with Figure 12, each electronic appliance 600 in the preferred embodiment includes one or more SPEs 503 and/or one or more <del>HPEs 655</del>. <del>These secure processing environments each provide a protected execution space for performing tasks in a secure manner.</del> They may fulfill service requests passed to them by ROS 602, and they may themselves generate service requests to be satisfied by other services within ROS 602 or by services provided by another VDE electronic appliance 600 or computer.</p> <p>In the preferred embodiment, an SPE 503 is supported by the hardware resources of an SPU 500. <del>An HPE 655 may be supported by general purpose processor resources and rely on software techniques for security/protection.</del> HPE 655 thus gives ROS 602 the capability of assembling and executing certain component assemblies 690 on a general purpose CPU such as a microcomputer, minicomputer, mainframe computer or supercomputer processor. In the preferred embodiment, the overall software architecture of an SPE 503 may be the same as the software architecture of an HPE 655. An HPE 655 can "emulate" SPE 503 and associated SPU 500, i.e., each may include services and resources needed to support an identical set of service requests from ROS 602 (although ROS 602</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>may be restricted from sending to an HPE certain highly secure tasks to be executed only within an SPU 500).</p> <p>'93 patent at 104:39-64; '900 patent at 112:2-27.</p> <hr/> <p><b>16(D)</b></p> <p>In the preferred embodiment, HPE 655 is a secure processing environment supported by a processor other than an SPU, such as for example an electronic appliance CPU 654 general-purpose microprocessor or other processing system or device. In the preferred embodiment, HPE 655 may be considered to "emulate" an SPU 500 in the sense that it may use software to provide some or all of the processing resources provided in hardware and/or firmware by an SPU. HPE 655 in one preferred embodiment of the present invention is full-featured and fully compatible with SPE 503—that is, HPE 655 can handle each and every service call SPE 503 can handle such that the SPE and the HPE are "plug compatible" from an outside interface standpoint (with the exception that the HPE may not provide as much security as the SPE).</p> <p>'93 patent at 79:60-80:7; '900 patent at 87:32-46.</p> <hr/> <p><b>16(E)</b></p> <p>Figure 12 also shows that ROS 602 may provide one or more SPEs 503 and/or one or more HPEs 655. As discussed above, HPE 655 may "emulate" an SPU 500 device, and such HPEs 655 may be integrated in lieu of (or in addition to) physical SPUs 500 for systems that need higher throughput. Some security may be lost since HPEs 655 are typically protected by operating system security and may not provide truly secure processing. Thus, in the preferred embodiment, for high security applications at least, all secure processing should take place within an SPE 503 having an execution space within a physical SPU 500 rather than a HPE 655 using software operating elsewhere in electronic appliance 600.</p> <p>'93 patent at 88:31-43; '900 patent at 96:6-18.</p> <hr/> <p><b>16(F)</b></p> <p>Occurrence of the control operation demonstrates that</p> |



|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>microprocessor 2652 is executing in its most privileged "normal" mode and therefore can be trusted to execute successfully the "enter 'SPU' mode" sequence of instructions stored in secure memory 532. <del>If microprocessor 2652 were not executing in its most privileged mode,</del> there would be no assurance that those instructions would execute successfully. Because switch 2663 isolates microprocessor 2652 from external signals (e.g., interrupts) until "SPU" mode is successfully initialized, the entry instructions can be guaranteed to complete successfully.</p> <p>'900 patent at 78:30-40.</p> <hr/> <p><b>16(G)</b></p> <p>Designing VDE capabilities into one or more standard microprocessor, microcontroller and/or other digital processing components may materially reduce VDE related hardware costs by employing the same hardware resources for both the transaction management uses contemplated by the present invention and for other, host electronic appliance functions. This means that a VDE SPU can employ (share) circuitry elements of a "standard" CPU. <del>For example, if a "standard" processor can operate in protected mode and can execute VDE related instructions as a protected activity, then such an embodiment may provide sufficient hardware security</del> for a variety of applications and the expense of a special purpose processor might be avoided. Under one preferred embodiment of the present invention, certain memory (e.g., RAM, ROM, NVRAM) is maintained during VDE related instruction processing in a protected mode (for example, as supported by protected mode microprocessors).</p> <p>'193 patent at 21:5-21; '900 patent at 21:1-17.</p> <hr/> <p><b>16(H)</b></p> <p>A VDE node's hardware SPU is a core component of a VDE secure subsystem and may employ some or all of an electronic appliance's primary control logic, such as a microcontroller, microcomputer or other CPU arrangement. This primary control logic may be otherwise employed for non VDE purposes such as the control of some or all of an electronic appliance's non-VDE functions. When operating in a hardware SPU mode, said primary control logic must be sufficiently secure so as to protect and conceal important VDE</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>processes. For example, a hardware SPU may employ a host electronic appliance microcomputer operating in protected mode while performing VDE related activities, thus allowing portions of VDE processes to execute with a certain degree of security. This alternate embodiment is in contrast to the preferred embodiment wherein a trusted environment is created using a combination of one or more tamper resistant semiconductors that are not part of said primary control logic.</p> <p>'193 patent at 49:33-50; '900 patent at 49:31-48.</p> |

|     | Claim Term / Phrase             | InterTrust Evidence   |
|-----|---------------------------------|---|
| 17. | identifier<br><br>193.15, 912.8 | <p><u>Patent Specifications</u></p> <p>17(A)</p> <p>This same termination (or other specified consequence such as budget reduction, price increase, message displays on screen to users, messages to administrators, etc.) can also be the consequence of the failure by a user or the users VDE installation to complete a monitored process, such as paying for usage in electronic currency, failure to perform backups of important stored information (e.g., content and/or appliance usage information, control information, etc.), failure to use a repeated failure to use the proper <u>passwords or other identifiers</u>, etc.).</p> <p>'193 patent at 270:12-21</p> <hr/> <p>17(B)</p> <p>During the same or different communication session, the terminal could similarly, securely communicate back to the portable appliance 2600 VDE secure subsystem details as to the retail transaction (for example, what was purchased and price, the retail establishment's digital signature, the <u>retail terminal's identifier</u>, tax related information, etc.).</p> <p>'193 patent at 233:35-41.</p> <hr/> <p>17(C)</p> <p>Many load modules 1100 contain code that executes in an SPE 503. Some load modules 1100 contain code that executes in an HPE 655. This allows methods 1000 to execute in whichever environment is appropriate. For example, an INFORMATION method 1000 can be built to execute only in SPE 503 secure space for government classes of security, or in an HPE 655 for commercial applications. As described above, the load module public header 802 may contain an "execution space code" field that indicates where the load module 1100 needs to execute. This functionality also allows for different SPE instruction sets as well as different user platforms, and allows methods to be constructed without dependencies on the underlying load module instruction set.</p> <p>'193 patent at 140:37-50.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>17(D)</p> <p>[VDE features] provide very <u>flexible and extensible user identification</u> according to individuals, installations, <u>by groups</u> such as classes, and by function and hierarchical identification employing a hierarchy of levels of client identification (for example, client organization ID, client department ID, client network ID, client project ID, and client employee ID, or any appropriate subset of the above).</p> <p>'193 patent at 25:31-38.</p> <hr/> <p>17(E)</p> <p>Account Numbers and User IDs</p> <p>In the preferred embodiment, to control access to clearinghouses, users are assigned account numbers at clearinghouses. Account numbers provide a unique "instance" value for a secure database record from the point of view of an outsider. From the point of view of an electronic appliance 600 site, the user, group, or group/user ids provide the unique instance of a record. For example, from the point of view of VISA, your Gold Card belongs to account number #123456789. From the point of view of the electronic appliance site (for example, a server at a corporation), the Gold card might belong to user id 1023. <u>In organizations which have plural users and/or user groups using a VDE node, such users and/or user groups will likely be assigned unique user IDs.</u></p> <p>'193 patent at 268:28-42.</p> <hr/> <p><u>Extrinsic Sources</u></p> <p>17(F)</p> <p><b>identify</b> v. <b>identified</b>, <b>identifying</b>, <b>identifies</b>. v. tr. 1. <u>To establish the identity of</u> 2. <u>To ascertain the origin, nature, or definitive characteristics of</u> 3. <i>Biology</i>. To determine the taxonomic classification of (an organism). 4. To consider as identical or united; equate. 5. To associate or affiliate (oneself) closely with a person or group. v. intr. To establish an identification with another or others. [Medieval Latin <i>identificare</i>, to make to resemble : Late Latin <i>identitas</i>, identity. See IDENTITY + Latin <i>-ficare</i>, <i>-fy</i>.]—<b>i-den'ti-fi'a-ble</b> <i>adj.</i> --<b>i-den'ti-fi'a-bly</b> <i>adv.</i> --<b>i-den'ti-fi'er</b> <i>n.</i></p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 896.</p> |

|     | Claim Term / Phrase                                   | InterTrust Evidence   |
|-----|---|---|
| 18. | protected processing environment<br><br>683.2, 721.34 | <p><b><u>Patent Specifications</u></b></p> <p><b>18(A)</b></p> <p>Because security may be better/more effectively enforced with the assistance of hardware security features such as those provided by SPU 500 (and because of other factors such as increased performance provided by special purpose circuitry within SPU 500), at least one SPE 503 is preferred for many or most higher security applications. However, in applications where lesser security can be tolerated and/or the cost of an SPU 500 cannot be tolerated, the SPE 503 may be omitted and all secure processing may instead be performed by one or more secure HPEs 655 executing on general-purpose CPUs 654.</p> <p>'193 patent 80:65-81:8.</p> <hr/> <p><b>18(B)</b></p> <p>The Ginter et al. patent disclosure describes, among other things, techniques for providing a secure, tamper resistant execution spaces within a "protected processing environment" for computer programs and data. The protected processing environment described in Ginter et al. may be hardware-based, software-based, or a hybrid.</p> <p>'721 patent 3:16-21.</p> <hr/> <p><b>18(C)</b></p> <p>One particular example of a secure execution space is a "protected processing environment" 108 of the type shown in Ginter et al. (see Figures 6-12) and described in associated text. Protected processing environments 108 provide a secure execution environment in which appliances 58, 60, 62 may securely execute load modules 54 to perform useful tasks.</p> <p>'721 patent 8:33-40.</p> <hr/> <p><b>18(D)</b></p> <p>In this example, appliance 600 may include one or more processors 4126 providing or supporting one or more "protected processing</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>environments" (PPE) 650 (e.g., SPEs 503 and/or HPEs 544) shown in Figures 6-12 and 62-72). Protected processing environment 650 may, for example, be implemented using a secure processing unit (SPU) 500 of the type shown in Figure 9 and/or may be based on software tamper-resistance techniques or a combination of software and hardware. As described above in detail, protected processing environment 650 provides a secure, trusted environment for storing, manipulating, executing, modifying and otherwise processing secure information such as that provided in secure electronic containers 302. In this particular example, secure containers 302 may not be opened except within a protected processing environment 650. Protected processing environment 650 is provided with the cryptographic and other information it needs to open and manipulate secure containers 302, and is tamper resistant so that an attacker cannot easily obtain and use this necessary information.</p> <p>'683 patent 29:51-30:3.</p> <hr/> <p><b>18(E)</b></p> <p>Figure 10 is a block diagram of one example of a software structure/architecture for Rights Operating System ("ROS") 602 provided by the preferred embodiment. In this example, ROS 602 includes an operating system ("OS") "core" 679, a user Application Program Interface ("API") 682, a "redirector" 684, an "intercept" 692, a User Notification/Exception Interface 686, and a file system 687. ROS 602 in this example also includes one or more Host Event Processing Environments ("HPEs") 655 and/or one or more Secure Event Processing Environments ("SPEs") 503 (these environments may be generically referred to as "Protected Processing Environments" 650).</p> <p>HPE(s) 655 and SPE(s) 503 are self-contained computing and processing environments that may include their own operating system kernel 688 including code and data processing resources. A given electronic appliance 600 may include any number of SPE(s) 503 and/or any number of HPE(s) 655. HPE(s) 655 and SPE(s) 503 may process information in a secure way, and provide secure processing support for ROS 602. For example, they may each perform secure processing based on one or more VDE component assemblies 690, and they may each offer secure processing services to OS kernel 680.</p> <p>In the preferred embodiment, SPE 503 is a secure processing environment provided at least in part by an SPU 500. Thus, SPU</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>500 provides the hardware tamper-resistant barrier 503 surrounding SPE 503. SPE 503 provided by the preferred embodiment is preferably:</p> <p>small and compact<br/>loadable into resource constrained environments such as for example minimally configured SPU 500<br/>dynamically updatable<br/>extensible by authorized users<br/>integratable into object or procedural environments<br/>secure.</p> <p>In the preferred embodiment, HPE 655 is a secure processing environment supported by a processor other than an SPU, such as for example an electronic appliance CPU 654 general-purpose microprocessor or other processing system or device. In the preferred embodiment, HPE 655 may be considered to "emulate" an SPU 500 in the sense that it may use software to provide some or all of the processing resources provided in hardware and/or firmware by an SPU. HPE 655 in one preferred embodiment of the present invention is full-featured and fully compatible with SPE 503—that is, HPE 655 can handle each and every service call SPE 503 can handle such that the SPE and the HPE are "plug compatible" from an outside interface standpoint (with the exception that the HPE may not provide as much security as the SPE).</p> <p>HPEs 655 may be provided in two types: secure and not secure. For example, it may be desirable to provide non-secure versions of HPE 655 to allow electronic appliance 600 to efficiently run non-sensitive VDE tasks using the full resources of a fast general purpose processor or computer. Such non-secure versions of HPE 655 may run under supervision of an instance of ROS 602 that also includes an SPE 503. In this way, ROS 602 may run all secure processes within SPE 503, and only use HPE 655 for processes that do not require security but that may require (or run more efficiently) under potentially greater resources provided by a general purpose computer or processor supporting HPE 655. Non-secure and secure HPE 655 may operate together with a secure SPE 503.</p> <p>'93 patent 79:24-80:21.</p> <hr/> <p>18(F)</p> <p>Figure 13 shows the software architecture of the preferred embodiment Secure Processing Environment (SPE) 503. This</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="623 338 1430 474">architecture may also apply to the preferred embodiment Host Processing Environment (HPE) 655. "Protected Processing Environment" ("PPE") 650 may refer generally to SPE 503 and/or HPE 655.</p> <p data-bbox="532 512 805 543">'93 patent 105:15-20.</p> <hr data-bbox="526 583 1458 590"/> <p data-bbox="532 625 602 657"><b>18(G)</b></p> <p data-bbox="618 695 1455 898">In some embodiments and where commercially acceptable, certain VDE participants, such as clearinghouses that normally maintain sufficiently physically secure non-VDE processing environments, may be allowed to employ HPEs rather VDE hardware elements and interoperate, for example, with VDE end-users and content providers.</p> <p data-bbox="532 936 786 968">'93 patent 13:17-23.</p> <hr data-bbox="526 1008 1458 1014"/> <p data-bbox="532 1052 602 1083"><b>18(H)</b></p> <p data-bbox="618 1121 1446 1430">An end user may make use of credit and/or currency securely stored within the end user's VDE installation secure subsystem to pay for charges related to use of VDE content received from the repository, and/or the user may maintain a secure credit and/or currency account remotely at the repository, including a "virtual" repository where payment is made for the receipt of such content by an end user. This later approach may provide greater assurance for payment to the repository and/or content providers particularly if the end user has only an HPE based secure subsystem.</p> <p data-bbox="532 1467 797 1499">'93 patent 291:39-49.</p> <hr data-bbox="526 1539 1458 1545"/> <p data-bbox="532 1583 586 1614"><b>18(I)</b></p> <p data-bbox="613 1652 1442 1921">One way to inexpensively and conveniently deploy multiple instances of VDE electronic appliances 600 across a network would be to provide network workstations with software defining an HPE 655. This arrangement requires no hardware modification of the workstations; an HPE 655 can be defined using software only. An SPE(s) 503 and/or HPE(s) 655 could also be provided within a VDE server. This arrangement has the advantage of allowing distributed VDE network processing without requiring workstations to be</p> |



|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>customized or modified (except for loading a new program(s) into them). VDE functions requiring high levels of security may be restricted to an SPU-based VDE server. "Secure" HPE-based workstations could perform VDE functions requiring less security, and could also coordinate their activities with the VDE server.</p> <p>'193 patent 226:43-57.</p> <hr/> <p><b>18(J)</b></p> <p>Large Organization Example</p> <p>In a somewhat more general example, suppose an organization (e.g., a corporation or government department) with thousands of employees and numerous offices disposed throughout a large geographic area wishes to exercise control over distribution of information which belongs to said organization (or association).</p> <p>'193 patent 277:26-32.</p> <hr/> <p><b>18(K)</b></p> <p>User Environment</p> <p>In an organization (or association) such as that described above, users may utilize a variety of electronic appliances 600 for processing and managing documents. This may include personal computers, both networked and otherwise, powerful single-user workstations, and servers or mainframe computers. To provide support for the control information described in this example, each electronic appliance that participates in use and management of VDE-protected documents may be enhanced with a VDE secure subsystem supporting an SPE 503 and/or HPE 655.</p> <p>In some organizations, where the threats to secure operation are relatively low, an HPE 655 may suffice. In other organizations (e.g., government defense), it may be necessary to employ an SPE 503 in all situations where VDE-protected documents are processed. The choice of enhancement environment and technology may be different in different of the organization. Even if different types of PPE 650 are used within an organization to serve different</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>requirements, they may be compatible and may operate on the same types (or subsets of types) of documents.</p> <p>'193 patent 278:45-65.</p> <hr/> <p><b>18(L)</b></p> <p>This manufacturing process may include, <del>for example</del>, testing the bootstrap loader and challenge-response software permanently stored within PPE 650, and <del>loading the PPE's unique ID</del>.</p> <p>'193 patent 223:36-39.</p> <hr/> <p><b>18(M)</b></p> <p><del>The level of security and tamper resistance required for trusted SPI hardware processes depends on the commercial requirements of particular markets or market niches, and may vary widely.</del></p> <p>'193 patent at 49:59-62.</p> <hr/> <p><b>18(N)</b></p> <p><del>There are many ways in which a PPE 650 might be compromised. The goal of the security provided by VDE 100 is to reduce the possibility that the system will be compromised, and minimize the adverse effects if it is compromised.</del></p> <p>'193 patent at 221:2-6.</p> <hr/> <p><b>18(O)</b></p> <p>VDE 100 provided by the preferred embodiment has <del>sufficient security to help ensure that it cannot be compromised short of a successful "brute force attack,"</del> and so that the time and cost to succeed in such a "brute force attack" substantially exceeds any value to be derived. In addition, the security provided by VDE 100 compartmentalizes the internal workings of VDE so that <del>a successful "brute force attack" would compromise only a strictly</del></p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="613 338 1382 369"><del>bounded subset of protected information, not the entire system.</del></p> <p data-bbox="524 411 829 443">'193 patent at 199:38-46.</p> <hr/> <p data-bbox="524 527 594 558"><b>18(P)</b></p> <p data-bbox="613 594 1409 699">VDE supports <del>trusted (sufficiently secure)</del> electronic information distribution and usage control models for both commercial electronic content distribution and data security applications.</p> <p data-bbox="524 737 813 768">'193 patent at 16:25-28.</p> <hr/> <p data-bbox="524 852 594 884"><b>18(Q)</b></p> <p data-bbox="613 919 1024 951">1. A security method comprising:</p> <p data-bbox="613 989 1438 1052">(a) digitally signing a first load module with a first digital signature designating the first load module for use by a first device class;</p> <p data-bbox="613 1089 1450 1293">(b) digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class;</p> <p data-bbox="613 1331 1450 1394">(c) distributing the first load module for use by at least one device in the first device class; and</p> <p data-bbox="613 1432 1365 1495">(d) distributing the second load module for use by at least one device in the second device class.</p> <p data-bbox="524 1541 797 1572">'721 patent at 21:9-24.</p> <hr/> <p data-bbox="524 1656 594 1688"><b>18(R)</b></p> <p data-bbox="613 1724 1252 1755">34. A protected processing environment comprising:</p> <p data-bbox="613 1793 1320 1824">a first tamper resistant barrier having a first security level,</p> <p data-bbox="613 1862 1029 1894">a first secure execution space, and</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.</p> <p>'721 patent at 24:48-56.</p> <hr/> <p><b>18(S)</b></p> <p>[VDE features] support security techniques that materially increase the time required to "break" a system's integrity. This includes using a collection of techniques that minimizes the damage resulting from comprising some aspect of the security features of the present inventions.</p> <p>'193 patent at 35:59-63.</p> <hr/> <p><b>18(T)</b></p> <p>Fingerprinting electronic content before it is encrypted for transfer to a customer or other user provides information that can be very useful for identifying who received certain content which may have then been distributed or made available in unencrypted form. This information would be useful in tracking who may have "broken" the security of a VDE installation and was illegally making certain electronic content available to others.</p> <p>'193 patent at 38:4-12.</p> <hr/> <p><b>18(U)</b></p> <p>If a content key becomes compromised, the portion of the content encrypted with the key is also compromised until the key "ages" and expires. If the "aging" process for that key also becomes compromised, then the breach becomes permanent.</p> <p>'193 patent at 222:49-53.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="532 338 607 369"><b>18(V)</b></p> <p data-bbox="623 407 1458 617">If PPE 650 (or a VDE administrator watching its activities or communicating with it) detects that it has been compromised, it may be updated with an initialization to use new code, keys and new encryption/decryption algorithms. This would limit exposure to VDE objects 300 that existed at the time the encryption scheme was broken.</p> <p data-bbox="532 653 824 684">'193 patent at 223:4-10.</p> <hr/> <p data-bbox="532 768 613 800"><b>18(W)</b></p> <p data-bbox="623 837 1435 1110">Secure VDE hardware (also known as SPU's for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host Processing Environments (HPEs)), operate in conjunction with secure communications, systems integration software, and distributed software control information and support structures, to achieve the electronic contract/rights protection environment of the present invention.</p> <p data-bbox="532 1146 808 1178">'193 patent at 13:7-14.</p> <hr/> <p data-bbox="532 1262 704 1293"><b><u>File Histories</u></b></p> <p data-bbox="532 1331 607 1362"><b>18(X)</b></p> <p data-bbox="623 1400 1451 1673">... the Examiner objects to the use of "environment" as indefinite and unclear. This word, however, is not used in isolation, but rather in the context of several longer phrases, all of which are defined in the specification. The phrase "protected processing environment," for example, is ... described on at least, for example, pages 7-8 and 25 of the specification. ... These terms are also described in the commonly assigned copending application ... filed 13 February 1995.</p> <p data-bbox="532 1709 1175 1740">'721 Patent File History, 4/13/99 Amendment, p. 13.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="527 327 1393 363"><u>Citations from Sources Designated by Microsoft under PLR 4-2(b)</u></p> <p data-bbox="527 396 602 432"><b>18(Y)</b></p> <p data-bbox="618 470 1425 535">Furthermore, there is never an absolute sense in which a system is secure or reliable.</p> <p data-bbox="527 573 1268 609">Neumann, Computer Related Risks (ACM Press, 1995), p. 2.</p> <hr/> <p data-bbox="527 686 597 722"><b>18(Z)</b></p> <p data-bbox="613 758 1442 997"><u>The fundamental purpose of security is to minimize the risk of loss from (1) physical damage or destruction, (2) human errors and omissions, and (3) theft or unauthorized disclosure. That purpose is best fulfilled by effective loss-prevention efforts. Loss-prevention efforts involve the identification and assessment of risks to capital, human, informational, and technological assets, and the development of suitable and cost-feasible countermeasures.</u></p> <p data-bbox="521 1035 1433 1071">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 75.</p> <hr/> <p data-bbox="521 1148 618 1184"><b>18(AA)</b></p> <p data-bbox="607 1220 1450 1808"><u>Total software security is no more attainable than is perfect security in any other area.</u> A highly skilled programmer can almost always penetrate software safeguards written by another programmer. Of course, the same can be said for attorneys; an unprincipled lawyer can usually get around protections in a contract written by an-other lawyer. Yet contracts continue to be written, and, for the most part, they are effective. Computer software security routines can also be effective most of the time. <u>A security procedure does not have to be all-encompassing, if it provides reasonable protection at an acceptable cost,</u> it is certainly worthwhile. The basic consideration is one of degree—how important are specific elements of data and software, and how important is their security. Some data require very little security. For example, a software library containing programs that are similar to those found in many other computer installations does not require elaborate security protection against theft. On the other hand, proprietary programs and sensitive data require extensive</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="618 338 1377 405">security. A data base containing payroll information requires stringent security procedures to maintain its confidentiality.</p> <p data-bbox="527 443 1455 478">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 201.</p> <hr/> <p data-bbox="527 554 618 590"><b>18(BB)</b></p> <p data-bbox="618 625 1422 730">Regardless of which form of data storage is being considered, one must bear in mind a vital concept, no data processing installation can afford 100 percent security—if indeed there is such a thing.</p> <p data-bbox="527 766 1455 802">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 218.</p> <hr/> <p data-bbox="527 877 618 913"><b>18(CC)</b></p> <p data-bbox="618 949 1393 1087">One hundred percent security cannot be achieved. The most effective systems apply security protection techniques in layers. Each layer of protection diminishes the chances of someone breaking through the barriers.</p> <p data-bbox="521 1123 1403 1190">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), pp. 293.</p> <hr/> <p data-bbox="521 1266 613 1302"><b>18(DD)</b></p> <p data-bbox="613 1337 1403 1509">Risk analysis is not intended to come up with a plan for absolute security. Indeed, absolute security is not achievable in today's computer's systems. Rather, risk analysis produces a degree of security commensurate with the information to be protected and with the amount of resources to be expended.</p> <p data-bbox="521 1545 1419 1612">Hoffman, Modern Methods for Computer Security and Privacy (Prentice-Hall, 1977), p. 170.</p> <hr/> <p data-bbox="521 1688 613 1724"><b>18(EF)</b></p> <p data-bbox="613 1759 1451 1934">No matter how secure you make them, computers can always be broken into given sufficient resources, time, and money. Computers are especially vulnerable because software is complex and we don't always know if there are flaws present that make the task of breaking in easier. Even systems that are certified according to the</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="618 331 1430 436">Department of Defense's so-called Orange Book are vulnerable, especially if they are not administered correctly. Just as six-foot-thick vaults doors don't work if they're not administered properly.</p> <p data-bbox="529 474 1446 541">Garfinkel et al., Practical Unix Security (O'Reilly &amp; Associates, 1991), pp. 13.</p> <hr data-bbox="529 583 1446 588"/> <p data-bbox="529 625 618 657"><b>18(FF)</b></p> <p data-bbox="618 695 1446 762"><del>Security is a relative, not an absolute, concept</del> and gains in security often come only with penalties in performance.</p> <p data-bbox="529 800 1446 867">Landwehr, Formal Models for Computer Security, ACM Computer Surveys (Sept. 3, 1981), p. 253.</p> |



|     | Claim Term / Phrase  | InterTrust Evidence  |
|-----|--|--|
| 19. | secure, securely<br><br>193.1, 193.11,<br>193.15, 861.58,<br>891.1, 683.2,<br>721.34, 912.8,<br>912.35 | <p><u>Patent Specifications</u></p> <p>19(A)</p> <p>VDE normally employs an integration of <u>cryptographic and other security technologies (e.g. encryption, digital signatures, etc.)</u>, with other technologies . . . .</p> <p>'193 patent 8:1-3.</p> <hr/> <p>19(B)</p> <p>Since VDE also employs <u>secure (e.g. encrypted and authenticated) communications</u> when passing information between the participant location (nodes) secure subsystems of a VDE arrangement, important components of a VDE electronic agreement can be reliably enforced with <u>sufficient security (sufficiently trusted) for the intended commercial purposes.</u></p> <p>'193 patent 45:39-45.</p> <hr/> <p>19(C)</p> <p><u>The degree of overall security of the VDE system is primarily dependent on the degree of tamper resistance and concealment of VDE control process execution and related data storage activities.</u></p> <p>'193 patent 21:26-29.</p> <hr/> <p>19(D)</p> <p>Because of the VDE <u>security, including use of effective encryption, authentication, digital signaturing, and secure database structures,</u> the records contained within a VDE card arrangement may be accepted as valid transaction records for government and/or corporate recordkeeping requirements.</p> <p>'193 patent 41:37-42.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="532 331 607 365"><b>19(E)</b></p> <p data-bbox="623 405 1468 785">SPU 500 is enclosed within and protected by a “tamper resistant security barrier” 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500. In one example, tamper resistant security barrier 502 is formed by security features such as encryption and hardware that detects tampering and/or destroys sensitive information within secure environment 503 when tampering is detected.</p> <p data-bbox="532 825 792 858">‘193 patent 59:48-59.</p> <hr/> <p data-bbox="532 936 607 970"><b>19(F)</b></p> <p data-bbox="623 1010 1419 1106">VDE 100 stores separately deliverable VDE elements in a secure (e.g., encrypted) database 610 distributed to each VDE electronic appliance 610.</p> <p data-bbox="532 1146 776 1180">‘193 patent 126:6-8.</p> <hr/> <p data-bbox="532 1257 607 1291"><b>19(G)</b></p> <p data-bbox="623 1331 1133 1362">Secure (tamper-resistant) executable code.</p> <p data-bbox="532 1402 802 1436">‘193 patent 126:30-31.</p> <hr/> <p data-bbox="532 1514 607 1547"><b>19(H)</b></p> <p data-bbox="623 1587 1419 1751">In one embodiment, the portable appliance 2600 could support secure (in this instance encrypted and/or authenticated) two-way communications with a retail terminal which may contain a VDE electronic appliance 600 or communicate with a retailer's or third party provider's VDE electronic appliance 600.</p> <p data-bbox="532 1791 802 1824">‘193 patent 233:25-30.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="532 323 597 359"><b>19(I)</b></p> <p data-bbox="621 394 1433 499">Information could then be automatically “parsed” and routed into <u>securely maintained (for example, encrypted)</u> appropriate database management records within portable appliance 2600.</p> <p data-bbox="532 533 808 569">‘193 patent 233:51-54.</p> <hr/> <p data-bbox="532 646 597 682"><b>19(J)</b></p> <p data-bbox="621 718 1450 823">The level of security and tamper resistance required for trusted SPU hardware processes depends on the commercial requirements of particular markets or market niches, and may vary widely.</p> <p data-bbox="532 856 816 892">‘193 patent at 49:59-62.</p> <hr/> <p data-bbox="532 970 605 1005"><b>19(K)</b></p> <p data-bbox="621 1041 1425 1178">There are many ways in which a PPE 650 might be compromised. The goal of the security provided by VDE 100 is to reduce the possibility that the system will be compromised, and minimize the adverse effects if it is compromised.</p> <p data-bbox="532 1211 800 1247">‘193 patent at 221:2-6.</p> <hr/> <p data-bbox="532 1325 597 1360"><b>19(L)</b></p> <p data-bbox="621 1396 1442 1671">VDE 100 provided by the preferred embodiment has <u>sufficient security to help ensure that it cannot be compromised short of a successful “brute force attack,”</u> and so that the time and cost to succeed in such a “brute force attack” substantially exceeds any value to be derived. In addition, the security provided by VDE 100 compartmentalizes the internal workings of VDE so that <u>a successful “brute force attack” would compromise only a strictly bounded subset of protected information, not the entire system.</u></p> <p data-bbox="532 1705 833 1740">‘193 patent at 199:38-46.</p> <hr/> <p data-bbox="532 1818 605 1854"><b>19(M)</b></p> <p data-bbox="621 1890 1409 1925">VDE supports <u>trusted (sufficiently secure)</u> electronic information</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>distribution and usage control models for both commercial electronic content distribution and data security applications.</p> <p>'193 patent at 16:25-28.</p> <hr/> <p><b>19(N)</b></p> <p>Because security may be better/more effectively enforced with the assistance of hardware security features such as those provided by SPU 500 (and because of other factors such as increased performance provided by special purpose circuitry within SPU 500), at least one SPE 503 is preferred for many or most higher security applications. However, in applications where lesser security can be tolerated and/or the cost of an SPU 500 cannot be tolerated, the SPE 503 may be omitted and all secure processing may instead be performed by one or more secure HPEs 655 executing on general-purpose CPUs 654.</p> <p>'193 patent at 80:65-81:8.</p> <hr/> <p><b>19(O)</b></p> <p>1. A security method comprising:</p> <p>(a) digitally signing a first load module with a first digital signature designating the first load module for use by a first device class;</p> <p>(b) digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class;</p> <p>(c) distributing the first load module for use by at least one device in the first device class; and</p> <p>(d) distributing the second load module for use by at least one device in the second device class.</p> <p>'721 patent at 21:9-24.</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="524 331 597 363"><b>19(P)</b></p> <p data-bbox="613 405 1445 783">34. A protected processing environment comprising:<br/><br/>a first tamper resistant barrier having a first security level,<br/><br/>a first secure execution space, and<br/><br/>at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.</p> <p data-bbox="524 825 824 856">'721 patent at 24:48-56.</p> <hr/> <p data-bbox="524 940 597 972"><b>19(Q)</b></p> <p data-bbox="613 1003 1445 1171">[VDE features] support security techniques that materially increase the time required to "break" a system's integrity. This includes using a collection of techniques that minimizes the damage resulting from comprising some aspect of the security features of the present inventions.</p> <p data-bbox="524 1213 824 1245">'193 patent at 35:59-63.</p> <hr/> <p data-bbox="524 1329 597 1360"><b>19(R)</b></p> <p data-bbox="613 1392 1445 1633">Fingerprinting electronic content before it is encrypted for transfer to a customer or other user provides information that can be very useful for identifying who received certain content which may have then been distributed or made available in unencrypted form. This information would be useful in tracking who may have "broken" the security of a VDE installation and was illegally making certain electronic content available to others.</p> <p data-bbox="524 1675 800 1707">'193 patent at 38:4-12.</p> <hr/> <p data-bbox="524 1791 597 1822"><b>19(S)</b></p> <p data-bbox="613 1854 1445 1917">If a content key becomes compromised, the portion of the content encrypted with the key is also compromised until the key "ages" and</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="613 327 1317 401">expires. If the "aging" process for that key also becomes compromised, then the breach becomes permanent.</p> <p data-bbox="521 432 834 468">'193 patent at 222:49-53.</p> <hr data-bbox="516 506 1463 512"/> <p data-bbox="521 548 597 583">19(T)</p> <p data-bbox="613 615 1451 825">If PPE 650 (or a VDE administrator watching its activities or communicating with it) detects that it has been compromised, it may be updated with an initialization to use new code, keys and new encryption/decryption algorithms. This would limit exposure to VDE objects 300 that existed at the time the encryption scheme was broken.</p> <p data-bbox="521 863 813 898">'193 patent at 223:4-10.</p> <hr data-bbox="516 936 1463 942"/> <p data-bbox="516 974 751 1010"><u>Extrinsic Sources</u></p> <p data-bbox="516 1041 597 1077">19(U)</p> <p data-bbox="613 1115 1422 1213"><b>security</b> The protection of valuable assets stored on computer systems or transmitted via computer networks. Computer security involves the following conceptually differentiated areas:</p> <ul data-bbox="656 1251 1422 1717" style="list-style-type: none"> <li>• <b>Authentication</b> (ensuring that users are indeed the persons they claim to be).</li> <li>• <b>Access control</b> (ensuring that users access only those resources and services that they are entitled to access).</li> <li>• <b>Confidentiality</b> (ensuring that transmitted or stored data is not examined by unauthorized persons).</li> <li>• <b>Integrity</b> (ensuring that transmitted or stored data is not altered by unauthorized persons in a way that is not detectable by authorized users).</li> <li>• <b>Nonrepudiation</b> (ensuring that qualified users are not denied access to services that they legitimately expect to receive, and that originators of messages cannot deny that they in fact sent a given message).</li> </ul> <p data-bbox="516 1749 1403 1818">Webster's New World Dictionary of Computer Terms, 6th ed. (1997), p. 463.</p> <hr data-bbox="516 1856 1463 1862"/> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="529 323 1390 359"><u>Citations from Sources Designated by Microsoft under PLR 4-2(b)</u></p> <p data-bbox="529 394 605 430"><b>19(V)</b></p> <p data-bbox="618 466 1442 569">In common technical usage, however, computer security and communication security generally refer to protection against human misuse, and exclude the protection against malfunctions.</p> <p data-bbox="529 604 1284 640">Neumann, Computer Related Risks (ACM Press, 1995), p. 96.</p> <hr/> <p data-bbox="529 718 610 753"><b>19(W)</b></p> <p data-bbox="618 789 1442 892">There is a fifth important attribute of dependability—the <i>security attribute</i>—that cannot be measured easily: the ability of a system to prevent unauthorized access or handling of information.</p> <p data-bbox="529 928 1422 963">Mullender, Distributed Systems, 2nd ed. (Addison-Wesley, 1993), p. 420.</p> <hr/> <p data-bbox="529 1041 600 1077"><b>19(X)</b></p> <p data-bbox="618 1113 1417 1173">Furthermore, there is never an absolute sense in which a system is secure or reliable.</p> <p data-bbox="529 1209 1263 1245">Neumann, Computer Related Risks (ACM Press, 1995), p. 2.</p> <hr/> <p data-bbox="529 1323 597 1358"><b>19(Y)</b></p> <p data-bbox="613 1394 1430 1633"><u>The fundamental purpose of security is to minimize the risk of loss from (1) physical damage or destruction, (2) human errors and omissions, and (3) theft or unauthorized disclosure. That purpose is best fulfilled by effective loss-prevention efforts. Loss-prevention efforts involve the identification and assessment of risks to capital, human, informational, and technological assets, and the development of suitable and cost-feasible countermeasures.</u></p> <p data-bbox="521 1669 1430 1705">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 75.</p> <hr/> <p data-bbox="521 1782 592 1818"><b>19(Z)</b></p> <p data-bbox="610 1854 1442 1921"><u>Total software security is no more attainable than is perfect security in any other area.</u> A highly skilled programmer can almost</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p>always penetrate software safeguards written by another programmer. Of course, the same can be said for attorneys; an unprincipled lawyer can usually get around protections in a contract written by an-other lawyer. Yet contracts continue to be written, and, for the most part, they are effective. Computer software security routines can also be effective most of the time. <u>A security procedure does not have to be all-encompassing, if it provides reasonable protection at an acceptable cost,</u> it is certainly worthwhile. The basic consideration is one of degree—how important are specific elements of data and software, and how important is their security. Some data require very little security. For example, a software library containing programs that are similar to those found in many other computer installations does not require elaborate security protection against theft. On the other hand, proprietary programs and sensitive data require extensive security. A data base containing payroll information requires stringent security procedures to maintain its confidentiality.</p> <p>Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 201.</p> <hr/> <p><b>19(AA)</b></p> <p><u>Regardless of which form of data storage is being considered, one must bear in mind a vital concept, no data processing installation can afford 100 percent security—if indeed there is such a thing.</u></p> <p>Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 218.</p> <hr/> <p><b>19(BB)</b></p> <p><u>One hundred percent security cannot be achieved.</u> The most effective systems apply security protection techniques in layers. Each layer of protection diminishes the chances of someone breaking through the barriers.</p> <p>Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), pp. 293.</p> <hr/> <p><b>19(CC)</b></p> <p>Risk analysis is not intended to come up with a plan for absolute security. Indeed, <u>absolute security is not achievable in today's</u></p> |



|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="618 331 1398 436"><del>computer's systems</del> Rather, risk analysis produces a degree of security commensurate with the information to be protected and with the amount of resources to be expended.</p> <p data-bbox="527 478 1422 541">Hoffman, Modern Methods for Computer Security and Privacy (Prentice-Hall, 1977), p. 170.</p> <hr data-bbox="527 583 1455 590"/> <p data-bbox="527 625 623 657"><b>19(DD)</b></p> <p data-bbox="618 695 1451 968"><del>No matter how secure you make them, computers can always be broken into given sufficient resources, time, and money.</del> Computers are especially vulnerable because software is complex and we don't always know if there are flaws present that make the task of breaking in easier. Even systems that are certified according to the Department of Defense's so-called Orange Book are vulnerable, especially if they are not administered correctly. Just as six-foot-thick vaults doors don't work if they're not administered properly.</p> <p data-bbox="527 1010 1435 1073">Garfinkel et al., Practical Unix Security (O'Reilly &amp; Associates, 1991), pp. 13.</p> <hr data-bbox="527 1115 1455 1121"/> <p data-bbox="527 1157 617 1188"><b>19(EF)</b></p> <p data-bbox="618 1226 1435 1289"><del>Security is a relative, not an absolute, concept</del> and gains in security often come only with penalties in performance.</p> <p data-bbox="527 1331 1451 1394">Landwehr, Formal Models for Computer Security, ACM Computer Surveys (Sept. 3, 1981), p. 253.</p> |

|     | Claim Term / Phrase  | InterTrust Evidence   |
|-----|--|---|
| 20. | <p data-bbox="277 327 488 359">secure container</p> <p data-bbox="277 411 467 474">912.35, 861.58, 683.2</p> | <p data-bbox="529 327 797 359"><u>Patent Specifications</u></p> <p data-bbox="529 401 602 432">20(A)</p> <p data-bbox="610 474 1445 1094"> The "container" concept is a convenient metaphor used to give a name to the collection of elements required to make use of content or to perform an administrative-type activity. Container 302 typically includes identifying information, control structures and content (e.g., a property or administrative data). The term "container" is often (e.g., Bento/OpenDoc and OLE) used to describe a collection of information stored on a computer system's secondary storage system(s) or accessible to a computer system over a communications network on a "server's" secondary storage system. The "container" 302 provided by the preferred embodiment is not so limited or restricted. In VDE 100, there is no requirement that this information is stored together, received at the same time, updated at the same time, used for only a single object, or be owned by the same entity. Rather, in VDE 100 the container concept is extended and generalized to include real-time content and/or online interactive content passed to an electronic appliance over a cable, by broadcast, or communicated by other electronic communication means. </p> <p data-bbox="521 1125 789 1157">'93 patent 127:30-49.</p> <hr/> <p data-bbox="521 1241 594 1272">20(B)</p> <p data-bbox="610 1314 1438 1871"> VDE, in its preferred embodiment, employs object software technology and uses object technology to form "containers" for delivery of information that is (at least in part) encrypted or otherwise secured. These containers may contain electronic content products or other electronic information and some or all of their associated permissions (control) information. These container objects may be distributed along pathways involving content providers and/or content users. They may be securely moved among nodes of a Virtual Distribution Environment (VDE) arrangement, which nodes operate VDE foundation software and execute control methods to enact electronic information usage control and/or administration models. The containers delivered through use of the preferred embodiment of the present invention may be employed both for distributing VDE control instructions (information) and/or to encapsulate and electronically distribute content that has been at least partially secured. </p> <p data-bbox="521 1902 789 1934">'93 patent 13:54-14:4.</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="526 338 602 369"><b>20(C)</b></p> <p data-bbox="618 411 1446 716">Figure 88 illustrates secure electronic container 302 as an attaché case handcuffed to the secure delivery person's wrist. Once again, container is shown as a physical thing for purposes of illustration only -- in the example it is preferably electronic rather than physical, and comprises digital information having a well-defined structure (see Figure 5A). Special mathematical techniques known as "cryptography" can be used to make electronic container 302 secure so that only intended recipient 4056 can open the container and access the electronic document (or other item) 4054 it contains.</p> <p data-bbox="526 758 805 789">'683 patent 15:61-16:4.</p> <hr/> <p data-bbox="526 873 602 905"><b>20(D)</b></p> <p data-bbox="618 947 1446 1314">The Figure 5A example shows items "within" and enclosed by container 302. However, container 302 may "contain" items without those items actually being stored within the container. For example, the container 302 may reference items that are available elsewhere such as in other containers at remote sites. Container 302 may reference items available at different times or only during limited times. Some items may be too large to store within container 302. Items may, for example, be delivered to the user in the form of a "live feed" of video at a certain time. Even then, the container 302 "contains" the live feed (by reference) in this example.</p> <p data-bbox="526 1356 781 1388">'193 patent 58:48-58.</p> <hr/> <p data-bbox="526 1472 602 1503"><b>20(E)</b></p> <p data-bbox="618 1545 1446 1913">The term "container" is often (e.g., Bento/OpenDoc and OLE) used to describe a collection of information stored on a computer system's secondary storage system(s) or accessible to a computer system over a communications network on a "server's" secondary storage system. The "container" 302 provided by the preferred embodiment is not so limited or restricted. In VDE 100, there is no requirement that this information is stored together, received at the same time, updated at the same time, used for only a single object or be owned by the same entity. Rather, in VDE 100 the container concept is extended and generalized to include real-time content and/or online interactive content passed to an electronic appliance.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="618 331 1393 401">over a cable, by broadcast, or communicated by other electronic communication means.</p> <p data-bbox="618 436 1446 856">Thus, the "complete" VDE container 302 or logical object structure 800 may not exist at the user's location (or any other location, for that matter) at any one time. The "logical object" may exist over a particular period of time (or periods of time), rather than all at once. This concept includes the notion of a "virtual container" where important container elements may exist either as a plurality of locations and/or over a sequence of time periods (which may or may not overlap). Of course, VDE 100 containers can also be stored with all required control structures and content together. This represents a continuum from all content and control structures present in a single container, to no locally accessible content or container-specific control structures.</p> <p data-bbox="524 892 802 926">'193 patent 127:35-62.</p> <hr/> <p data-bbox="524 968 591 1001"><b>20(F)</b></p> <p data-bbox="613 1037 1393 1136">In order to improve performance, the containers themselves may remain at the users' sites, and only the encrypted contents transmitted between the participants.</p> <p data-bbox="524 1178 753 1211">'683 patent 53:3-5.</p> <hr/> <p data-bbox="524 1289 591 1323"><b>20(G)</b></p> <p data-bbox="613 1358 1438 1707">In more detail, the logical object structure 800 provided by the preferred embodiment includes a public (or unencrypted) header 802 that identifies the object and may also identify one or more owners of rights in the object and/or one or more distributors of the object. Private (or encrypted) header 804 may include a part or all of the information in the public header and further, in the preferred embodiment, will include additional data for validating and identifying the object 300 when a user attempts to register as a user of the object with a service clearinghouse, VDE administrator, or an SPU 500. Alternatively, information identifying....</p> <p data-bbox="524 1743 802 1776">'193 patent 128:11-21.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="521 331 597 363"><b>20(H)</b></p> <p data-bbox="613 401 1406 468">Third party go-between can authenticate an item by, for example, <u>opening (e.g. decrypting content)</u> one or more containers</p> <p data-bbox="521 506 768 537">'683 patent 9:59-61.</p> <hr data-bbox="516 577 1451 583"/> <p data-bbox="521 621 748 653"><b><u>Extrinsic Sources</u></b></p> <p data-bbox="521 690 586 722"><b>20(I)</b></p> <p data-bbox="613 760 1430 827"><b>container</b> <i>n.</i> 1. In OLE terminology, <u>a file containing linked or embedded objects</u>. See also OLE. 2. In SGML, an element that has</p> <p data-bbox="613 865 1382 932">content as opposed to one consisting solely of the tag name and attributes.</p> <p data-bbox="521 970 1393 1001">Microsoft Computer Dictionary, 3d, ed. (Microsoft Press, 1997), p. 115.</p> <hr data-bbox="516 1039 1451 1045"/> <p data-bbox="521 1083 586 1115"><b>20(J)</b></p> <p data-bbox="613 1152 1438 1633">In a preferred embodiment of the present invention, an application program that creates a compound document controls the manipulation of linked or embedded data generated by another application. In object-oriented parlance, this data is referred to as an object. (The reference Budd, T., "An Introduction to Object-Oriented Programming," Addison-Wesley Publishing Co., Inc., 1991, provides an introduction to object-oriented concepts and terminology.) <u>An object that is either linked or embedded into a compound document is "contained" within the document. Also, a compound document is referred to as a "container" object and the objects contained within a compound document are referred to as "contained" or "containee" objects. Referring to FIGS. 1 and 2, the scheduling data 102 and budgeting data 103 are containee objects and the compound document 101 is a container object.</u></p> <p data-bbox="521 1671 841 1703">USP 5,634,019 at 7:34-49.</p> |

|     | Claim Term / Phrase            | InterTrust Evidence  |
|-----|--------------------------------|--|
| 21. | tamper resistance<br><br>721.1 | <p><u>Patent Specifications</u></p> <p>21(A)</p> <p>Maintaining shared secrets (e.g., cryptographic keys) within a tamper resistant enclosure that the owner of the electronic appliance cannot easily tamper with.</p> <p>'721 patent at 4:40-42.</p> <hr/> <p>21(B)</p> <p>SPU 500 is enclosed within and protected by a "tamper resistant security barrier" 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500. In one example, tamper resistant security barrier 502 is formed by security features such as "encryption," and hardware that detects tampering and/or destroys sensitive information within secure environment 503 when tampering is detected.</p> <p>'193 patent at 59:48-59.</p> <hr/> <p><u>Extrinsic Sources</u></p> <p>21(C)</p> <p>To evaluate the results of physically protecting portions of the system, the concept of a tamper-resistant module (TRM) is introduced. All information contained within a TRM is protected from disclosure and undetected modification in the following sense. As long as the TRM is intact, data inside the module cannot be discerned or modified by an attacker and if the TRM is breached the sensitive data within is destroyed (erased). The implementation of TRMs will vary considerably depending on the value of the external software being protected and the perceived sophistication of potential attackers.</p> <p>Kent, Protecting Externally Supplied Software in Small Computers, Doctoral Thesis (Sept. 22, 1980), p. PA00000363.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="521 338 597 369"><b>21(D)</b></p> <p data-bbox="613 407 1446 617"> <u>Tamper resistant software is software which is resistant to observation and modification.</u> It can be trusted/ within certain bounds/ to operate as intended even in the presence of a malicious attack. Our approach has been to classify attacks into three categories and then to develop a series of software design principles that allow a scaled response to those threats. </p> <p data-bbox="521 653 1365 716">Aucsmith, Tamper Resistant Software: An Implementation (1996), p. PA00002323.</p> <hr data-bbox="516 758 1453 764"/> <p data-bbox="521 800 597 831"><b>21(E)</b></p> <p data-bbox="613 869 1406 1005"> <u>Tamper-resistance ensures proper operation of a program and prevents extraction of secret data and abuse of the program.</u> Moreover tamper-resistance enables a vendor to enforce his own conditions upon users. </p> <p data-bbox="521 1043 1390 1180">Mambo et al., A Tentative Approach to Constructing Tamper-Resistant Software, School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai Tatsunokuchi Nomi, Ishikawa (1997), p. PA00005363.</p> |

|     | Claim Term / Phrase                    | InterTrust Evidence  |
|-----|--|--|
| 22. | tamper resistant barrier<br><br>721.34 | <p><u>Patent Specifications</u></p> <p>22(A)</p> <p>SPU 500 is enclosed within and protected by a “tamper resistant security barrier” 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500. In one example, tamper resistant security barrier 502 is formed by security features such as “encryption,” and hardware that detects tampering and/or destroys sensitive information within secure environment 503 when tampering is detected.</p> <p>‘193 patent 59:48-59.</p> <hr/> <p>22(B)</p> <p>HPEs 655 may (as shown in Figure 10) be provided with a <del>software-based tamper resistant barrier</del> 674 that makes them more secure. Such a software-based tamper resistant barrier 674 may be created by software executing on general-purpose CPU 654. Such a “secure” HPE 655 can be used by ROS 602 to execute processes that, while still needing security, may not require the degree of security provided by SPU 500. This can be especially beneficial in architectures providing both an SPE 503 and an HPE 655. The SPU 502 may be used to perform all truly secure processing, whereas one or more HPEs 655 may be used to provide additional secure (albeit possibly less secure than the SPE) processing using host processor or other general purpose resources that may be available within an electronic appliance 600. Any service may be provided by such a secure HPE 655. In the preferred embodiment, certain aspects of “channel processing” appears to be a candidate that could be readily exported from SPE 503 to HPE 655.</p> <p>The software-based tamper resistant barrier 674 provided by HPE 655 may be provided, for example, by: introducing time checks and/or code modifications to complicate the process of stepping through code comprising a portion of kernel 688a and/or a portion of component assemblies 690 using a debugger; using a map of defects on a storage device (e.g., a hard disk, memory card, etc.) to form internal test values to impede moving and/or copying HPE 655 to other electronic appliances 600; using kernel code that contains</p> |



|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>false branches and other complications in flow of control to disguise internal processes to some degree from disassembly or other efforts to discover details of processes; using “self-generating” code (based on the output of a co-sine transform, for example) such that detailed and/or complete instruction sequences are not stored explicitly on storage devices and/or in active memory but rather are generated as needed; using code that “shuffles” memory locations used for data values based on operational parameters to complicate efforts to manipulate such values; using any software and/or hardware memory management resources of electronic appliance 600 to “protect” the operation of HPE 655 from other processes, functions, etc. Although such a software-based tamper-resistant barrier 674 may provide a fair degree of security, it typically will not be as secure as the hardware-based tamper-resistant barrier 502 provided (at least in part) by SPU 500.</p> <p>‘193 patent 80:22-65.</p> <hr/> <p>22(C)</p> <p>Protected execution spaces such as protected processing environments can be programmed or otherwise conditioned to accept only those load modules or other executables bearing a digital signature/certificate of an accredited (or particular) verifying authority. Tamper-resistant barriers may be used to protect this programming or other conditioning.</p> <p>‘721 patent 5:1-6.</p> |

|     | Claim Term /<br>Phrase   | InterTrust Evidence  |
|-----|--|--|
| 23. | <p>use</p> <p>912.8, 912.35,<br/>861.58, 193.19,<br/>891.1, 683.2,<br/>721.1</p> | <p><u>Extrinsic Sources</u></p> <p>23(A)</p> <p><b>use v. used, us-ing, us-es. tr.</b> 1. <del>To put into service or apply for a purpose; employ.</del> 2. To avail oneself of; practice: <i>use caution</i>. 3. To conduct oneself toward; treat or handle: "<i>the peace offering of a man who once used you unkindly</i>" (Laurence Sterne). 4. To seek or achieve an end by means of; exploit: <i>used their highly placed friends to gain access to the president; felt he was being used by seekers of favor</i>. 5. To take or consume; partake of: <i>She rarely used alcohol.</i> –intr. (yoos, yoost). Used in the past tense followed by <i>to</i> in order to indicate a former state, habitual practice, or custom: <i>Mail service used to be faster.</i> <b>use</b> (yoos). <b>n.</b> 1. <b>a.</b> The act of using; the application or employment of something for a purpose: <i>with the use of a calculator; skilled in the use of the bow and arrow.</i> <b>b.</b> The condition or fact of being used: <i>a chair in regular use.</i> 2. The manner of using; usage: <i>learned the proper use of power tools.</i> 3. <b>a.</b> The permission, privilege, or benefit of using something: <i>gave us the use of their summerhouse.</i> <b>b.</b> The power or ability to use something: <i>lost the use of one arm.</i> 4. The need or occasion to use or employ: <i>have no use for these old clothes.</i> 5. The quality of being suitable or adaptable to an end; usefulness: <i>tried to be of use in the kitchen.</i> 6. A purpose for which something is used: <i>a tool with several uses; a pretty bowl, but of what use is it?</i> 7. Gain or advantage; good: <i>There's no use in discussing it. What's the use?</i> 8. Accustomed or usual procedure or practice. 9. <b>Law.</b> <b>a.</b> Enjoyment of property, as by occupying or exercising it. <b>b.</b> The benefit or profit of lands and tenements of which the legal title and possession are vested in another. <b>c.</b> The arrangement establishing the equitable right to such benefits and profits. 10. A liturgical form practiced in a particular church, ecclesiastical district, or community. 11. <i>Obsolete.</i> Usual occurrence or experience. --<b>phrasal verb.</b> <b>use up.</b> To consume completely: <i>used up all our money.</i> [Middle English <i>usen</i>, from Old French <i>user</i>, from Vulgar Latin <i>*usare</i>, frequentative of Latin <i>uti</i>.]</p> <p><b>SYNONYM:</b> <i>use, employ, utilize.</i> These verbs mean to avail oneself of someone or something in order to make him, her, or it useful, functional, or beneficial. To <i>use</i> is to put into service or apply for a purpose: <i>uses a hearing aid; used the press secretary as spokesperson for the administration; using a stick to stir the paint.</i> <i>Employ</i> is often interchangeable with <i>use</i>: <i>She employed her education to maximum advantage.</i> Unlike <i>use</i>, however, the term can denote engaging or maintaining the services of another or putting another to work: "<i>When men are employed, they are best</i></p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p><i>contented"</i> (Benjamin Franklin). <i>Utilize</i> is especially appropriate in the narrower sense of making something profitable or of finding new and practical uses for it: <i>In the 19th century waterpower was widely utilized to generate electricity</i>. See also Synonyms at <b>habit</b>.</p> <p>American Heritage Dictionary, 3d ed. (Houghton Mifflin, 1992), p. 1966.</p> |

|     | Claim Term / Phrase                             | InterTrust Evidence  |
|-----|---|--|
| 24. | virtual distribution environment<br><br>900.155 | <p><u>Patent Specifications</u></p> <p><b>24(A)</b></p> <p>VDE provides a secure, distributed electronic transaction management system for controlling the distribution and/or other usage of electronically provided and/or stored information.</p> <p>'193 patent at 9:36-39; '900 patent at 9:33-36.</p> <hr/> <p><b>24(B)</b></p> <p>Electronic appliances such as computers employed in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Secure subsystems used with such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions.</p> <p>'900 patent at Abstract.</p> <hr/> <p><b>24(C)</b></p> <p>Figure 1 shows a "Virtual Distribution Environment" ("VDE") 100 that may be provided in accordance with this invention. In Figure 1, an information utility 200 connects to communications means 202 such as telephone or cable TV lines for example. Telephone or cable TV lines 202 may be part of an "electronic highway" that carries electronic information from place to place. Lines 202 connect information utility 200 to other people such as for example a consumer 208, an office 210, a video production studio 204, and a publishing house 214. Each of the people connected to information utility 200 may be called a "VDE participant" because they can participate in transactions occurring within the virtual distribution environment 100.</p> <p>Almost any sort of transaction you can think of can be supported by virtual distribution environment 100. A few of many examples of</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>transactions that can be supported by virtual distribution environment 100 include:</p> <p>home banking and electronic payments;</p> <p>electronic legal contracts;</p> <p>distribution of "content" such as electronic printed matter, video, audio, images and computer programs; and</p> <p>secure communication of private information such as medical records and financial information.</p> <p>Virtual distribution environment 100 is "virtual" because it does not require many of the physical "things" that used to be necessary to protect rights, ensure reliable and predictable distribution, and ensure proper compensation to content creators and distributors. For example, in the past, information was distributed on records or disks that were difficult to copy. In the past, private or secret content was distributed in sealed envelopes or locked briefcases delivered by courier. To ensure appropriate compensation, consumers received goods and services only after they handed cash over to a seller. Although information utility 200 may deliver information by transferring physical "things" such as electronic storage media, the virtual distribution environment 100 facilitates a completely electronic "chain of handling and control."</p> <p>'193 patent at 52:66-53:37; '900 patent 53:39-54:10.</p> <hr/> <p><b>24(D)</b></p> <p>Because security may be better/more effectively enforced with the assistance of hardware security features such as those provided by SPU 500 (and because of other factors such as increased performance provided by special purpose circuitry within SPU 500), at least one SPE 503 is preferred for many or most higher security applications. However, in applications where lesser security can be tolerated and/or the cost of an SPU 500 cannot be tolerated, the SPE 503 may be omitted and all secure processing may instead be performed by one or more secure HPEs 655 executing on general-purpose CPUs 654.</p> <p>'193 patent 80:65-67-81:8.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="526 331 607 363"><b>24(E)</b></p> <p data-bbox="618 405 1451 716">An end user may make use of credit and/or currency securely stored within the end user's VDE installation secure subsystem to pay for charges related to use of VDE content received from the repository, and/or the user may maintain a secure credit and/or currency account remotely at the repository, including a "virtual" repository where payment is made for the receipt of such content by an end user. This later approach may provide greater assurance for payment to the repository and/or content providers <u>particularly if the end user has only an HPE-based secure subsystem.</u></p> <p data-bbox="526 751 1118 783">'193 patent at 291:39-49; '900 patent 316:35-45.</p> <hr data-bbox="526 825 1451 829"/> <p data-bbox="526 867 607 898"><b>24(F)</b></p> <p data-bbox="618 936 971 968">Large Organization Example</p> <p data-bbox="618 1005 1451 1173">In a somewhat more general example, suppose an organization (e.g., a corporation or government department) with thousands of employees and numerous offices disposed throughout a large geographic area wishes to exercise control over distribution of information which belongs to said organization (or association).</p> <p data-bbox="526 1209 1118 1241">'193 patent at 277:26-32; '900 patent 302:17-24.</p> <hr data-bbox="526 1283 1451 1287"/> <p data-bbox="526 1325 607 1356"><b>24(G)</b></p> <p data-bbox="618 1394 841 1425">User Environment</p> <p data-bbox="618 1463 1430 1774">In an organization (or association) such as that described above, users may utilize a variety of electronic appliances 600 for processing and managing documents. This may include personal computers, both networked and otherwise, powerful single-user workstations, and servers or mainframe computers. To provide support for the control information described in this example, each electronic appliance that participates in use and management of VDE-protected documents may be enhanced with a <u>VDE secure subsystem supporting an SPE 503 and/or HPE 655.</u></p> <p data-bbox="618 1812 1419 1913"><u>In some organizations, where the threats to secure operation are relatively low, an HPE 655 may suffice. In other organizations (e.g., government defense), it may be necessary to employ an SPE</u></p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="618 327 1446 537">503 in all situations where VDE-protected documents are processed. The choice of enhancement environment and technology may be different in different of the organization. Even if different types of PPE 650 are used within an organization to serve different requirements, they may be compatible and may operate on the same types (or subsets of types) of documents.</p> <p data-bbox="529 575 1117 606">'93 patent at 278:45-65; '900 patent 303:40-61.</p> <hr data-bbox="521 646 1455 653"/> <p data-bbox="526 688 602 720"><b>24(H)</b></p> <p data-bbox="613 758 1446 1308">HPEs 655 may (as shown in Figure 10) be provided with a software-based tamper resistant barrier 674 that makes them more secure. Such a software-based tamper resistant barrier 674 may be created by software executing on general-purpose CPU 654. Such a "secure" HPE 655 can be used by ROS 602 to execute processes that, while still needing security, may not require the degree of security provided by SPU 500. This can be especially beneficial in architectures providing both an SPE 503 and an HPE 655. The SPU 502 may be used to perform all truly secure processing, whereas one or more HPEs 655 may be used to provide additional secure (albeit possibly less secure than the SPE) processing using host processor or other general purpose resources that may be available within an electronic appliance 600. Any service may be provided by such a secure HPE 655. In the preferred embodiment, certain aspects of "channel processing" appears to be a candidate that could be readily exported from SPE 503 to HPE 655.</p> <p data-bbox="610 1346 1446 1932">The software-based tamper resistant barrier 674 provided by HPE 655 may be provided, for example, by: introducing time checks and/or code modifications to complicate the process of stepping through code comprising a portion of kernel 688a and/or a portion of component assemblies 690 using a debugger; using a map of defects on a storage device (e.g., a hard disk, memory card, etc.) to form internal test values to impede moving and/or copying HPE 655 to other electronic appliances 600; using kernel code that contains false branches and other complications in flow of control to disguise internal processes to some degree from disassembly or other efforts to discover details of processes; using "self-generating" code (based on the output of a co-sine transform, for example) such that detailed and/or complete instruction sequences are not stored explicitly on storage devices and/or in active memory but rather are generated as needed; using code that "shuffles" memory locations used for data values based on operational parameters to complicate efforts to manipulate such values; using any software and/or hardware</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>memory management resources of electronic appliance 600 to “protect” the operation of HPE 655 from other processes, functions, etc. Although such a software-based tamper resistant barrier 674 may provide a fair degree of security, it typically will not be as secure as the hardware-based tamper resistant barrier 502 provided (at least in part) by SPU 500.</p> <p>‘193 patent 80:22-65.</p> <hr/> <p><b>24(I)</b></p> <p>VDE supplies an efficient, largely transparent, low cost and sufficiently secure system (supporting both hardware/software and software only models).</p> <p>‘193 patent at 9:11-13; ‘900 patent 9:8-10.</p> <hr/> <p><b>24(J)</b></p> <p>10. A method as in claim 1 in which said steps of receiving, providing, performing and producing occur within a Virtual Distribution Environment.</p> <p>11. A system as in claim 2 in which said first location and said second location are contained within a Virtual Distribution Environment.</p> <p>12. A system as in claim 3 in which said first location and said second location are contained within a Virtual Distribution Environment.</p> <p>13. A system as in claim 6 in which said protected processing environment is contained within a Virtual Distribution Environment.</p> <p>14. A method as in claim 9 in which said first location and said second location are contained within a Virtual Distribution Environment.</p> <p>USP 5,949,876 at 320:14-28.</p> <hr/> |



|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="526 327 602 359"><b>24(K)</b></p> <p data-bbox="618 394 1442 499">The level of security and tamper resistance required for trusted SPU hardware processes depends on the commercial requirements of particular markets or market niches, and may vary widely.</p> <p data-bbox="526 537 818 569">'193 patent at 49:59-62.</p> <hr/> <p data-bbox="526 653 602 684"><b>24(L)</b></p> <p data-bbox="618 720 1422 856">There are many ways in which a PPE 650 might be compromised. The goal of the security provided by VDE 100 is to reduce the possibility that the system will be compromised, and minimize the adverse effects if it is compromised.</p> <p data-bbox="526 894 802 926">'193 patent at 221:2-6.</p> <hr/> <p data-bbox="526 1010 602 1041"><b>24(M)</b></p> <p data-bbox="618 1077 1433 1350">VDE 100 provided by the preferred embodiment has sufficient security to help ensure that it cannot be compromised short of a successful "brute force attack," and so that the time and cost to succeed in such a "brute force attack" substantially exceeds any value to be derived. In addition, the security provided by VDE 100 compartmentalizes the internal workings of VDE so that a successful "brute force attack" would compromise only a strictly bounded subset of protected information, not the entire system.</p> <p data-bbox="526 1388 829 1419">'193 patent at 199:38-46.</p> <hr/> <p data-bbox="526 1503 602 1535"><b>24(N)</b></p> <p data-bbox="618 1570 1414 1675">VDE supports trusted (sufficiently secure) electronic information distribution and usage control models for both commercial electronic content distribution and data security applications.</p> <p data-bbox="526 1713 813 1745">'193 patent at 16:25-28.</p> <hr/> <p data-bbox="526 1829 602 1860"><b>24(O)</b></p> <p data-bbox="618 1896 1203 1927">Employing VDE as a general purpose electronic</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="613 321 1442 636">transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model.</p> <p data-bbox="521 678 1109 709">'193 patent at 11:38-49; '900 patent at 11:36-47.</p> <hr data-bbox="516 751 1446 756"/> <p data-bbox="521 793 589 825"><b>24(P)</b></p> <p data-bbox="613 856 1442 1024">[VDE features] support security techniques that materially increase the time required to "break" a system's integrity. This includes using a collection of techniques that minimizes the damage resulting from comprising some aspect of the security features of the present inventions.</p> <p data-bbox="521 1066 800 1098">'193 patent at 35:59-63</p> <hr data-bbox="516 1140 1446 1144"/> <p data-bbox="521 1182 589 1213"><b>24(Q)</b></p> <p data-bbox="613 1245 1442 1486">Fingerprinting electronic content before it is encrypted for transfer to a customer or other user provides information that can be very useful for identifying who received certain content which may have then been distributed or made available in unencrypted form. This information would be useful in tracking who may have "broken" the security of a VDE installation and was illegally making certain electronic content available to others.</p> <p data-bbox="521 1528 784 1560">'193 patent at 38:4-12</p> <hr data-bbox="516 1602 1446 1606"/> <p data-bbox="521 1644 589 1675"><b>24(R)</b></p> <p data-bbox="613 1707 1442 1843">If a content key becomes compromised, the portion of the content encrypted with the key is also compromised until the key "ages" and expires. If the "aging" process for that key also becomes compromised, then the breach becomes permanent.</p> <p data-bbox="521 1885 816 1917">'193 patent at 222:49-53.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="524 327 594 359">24(S)</p> <p data-bbox="613 394 1446 604">If PPE 650 (or a VDE administrator watching its activities or communicating with it) detects that it has been compromised, it may be updated with an initialization to use new code, keys and new encryption/decryption algorithms. This would limit exposure to VDE objects 300 that existed at the time the encryption scheme was broken.</p> <p data-bbox="524 642 813 674">'93 patent at 223:4-10.</p> <hr/> <p data-bbox="524 758 1385 789"><u>Citations from Sources Designated by Microsoft under PLR 4-2(b)</u></p> <p data-bbox="524 827 594 858">24(T)</p> <p data-bbox="613 894 1414 961">Furthermore, there is never an absolute sense in which a system is secure or reliable.</p> <p data-bbox="524 999 1263 1031">Neumann, Computer Related Risks (ACM Press, 1995), p. 2.</p> <hr/> <p data-bbox="524 1110 594 1142">24(U)</p> <p data-bbox="613 1178 1430 1419">The fundamental purpose of security is to minimize the risk of loss from (1) physical damage or destruction, (2) human errors and omissions, and (3) theft or unauthorized disclosure. That purpose is best fulfilled by effective loss-prevention efforts. Loss-prevention efforts involve the identification and assessment of risks to capital, human, informational, and technological assets, and the development of suitable and cost-feasible countermeasures.</p> <p data-bbox="524 1457 1430 1488">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 75.</p> <hr/> <p data-bbox="524 1568 594 1600">24(V)</p> <p data-bbox="613 1635 1446 1913">Total software security is no more attainable than is perfect security in any other area. A highly skilled programmer can almost always penetrate software safeguards written by another programmer. Of course, the same can be said for attorneys; an unprincipled lawyer can usually get around protections in a contract written by an-other lawyer. Yet contracts continue to be written, and, for the most part, they are effective. Computer software security routines can also be effective most of the time. A</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="618 331 1446 716"> <del>security procedure does not have to be all-encompassing; if it provides reasonable protection at an acceptable cost,</del> it is certainly worthwhile. The basic consideration is one of degree—how important are specific elements of data and software, and how important is their security. Some data require very little security. For example, a software library containing programs that are similar to those found in many other computer installations does not require elaborate security protection against theft. On the other hand, proprietary programs and sensitive data require extensive security. A data base containing payroll information requires stringent security procedures to maintain its confidentiality. </p> <p data-bbox="524 751 1446 785">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 201.</p> <hr/> <p data-bbox="524 867 610 900">24(W)</p> <p data-bbox="618 936 1414 1037"> <del>Regardless of which form of data storage is being considered, one must bear in mind a vital concept: no data processing installation can afford 100 percent security—if indeed there is such a thing.</del> </p> <p data-bbox="524 1073 1446 1106">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), p. 218.</p> <hr/> <p data-bbox="524 1188 600 1222">24(X)</p> <p data-bbox="618 1257 1390 1392"> <del>One hundred percent security cannot be achieved.</del> The most effective systems apply security protection techniques in layers. Each layer of protection diminishes the chances of someone breaking through the barriers. </p> <p data-bbox="524 1428 1398 1493">Hutt et al., Computer Security Handbook, 2d ed. (Macmillan, 1988), pp. 293.</p> <hr/> <p data-bbox="524 1575 597 1608">24(Y)</p> <p data-bbox="618 1644 1398 1818"> Risk analysis is not intended to come up with a plan for absolute security. Indeed, <del>absolute security is not achievable in today's computer systems.</del> Rather, risk analysis produces a degree of security commensurate with the information to be protected and with the amount of resources to be expended. </p> <p data-bbox="524 1854 1414 1919">Hoffman, Modern Methods for Computer Security and Privacy (Prentice-Hall, 1977), p. 170.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="526 327 602 359"><b>24(Z)</b></p> <p data-bbox="618 401 1446 674"> <u>No matter how secure you make them, computers can always be broken into given sufficient resources, time, and money.</u> Computers are especially vulnerable because software is complex and we don't always know if there are flaws present that make the task of breaking in easier. Even systems that are certified according to the Department of Defense's so-called Orange Book are vulnerable, especially if they are not administered correctly. Just as six-foot-thick vaults doors don't work if they're not administered properly. </p> <p data-bbox="526 716 1430 779">Garfinkel et al., Practical Unix Security (O'Reilly &amp; Associates, 1991), pp. 13.</p> <hr/> <p data-bbox="526 863 618 894"><b>24(AA)</b></p> <p data-bbox="618 936 1430 999"> <u>Security is a relative, not an absolute, concept,</u> and gains in security often come only with penalties in performance. </p> <p data-bbox="526 1041 1446 1104">Landwehr, Formal Models for Computer Security, ACM Computer Surveys (Sept. 3, 1981), p. 253.</p> <hr/> <p data-bbox="526 1188 691 1220"><b><u>File Histories</u></b></p> <p data-bbox="526 1251 618 1283"><b>24(BB)</b></p> <p data-bbox="618 1325 1414 1388">1. Restriction to <u>one of the following inventions</u> is required under 35 U.S.C. § 121:</p> <p data-bbox="618 1430 1414 1493">Group I . . . drawn to a secure component-based operating process, classified in Classs 380, subclass 25.</p> <p data-bbox="618 1535 1349 1598">Group II. . . . drawn to method(s) for managing a resource or operating, classified in Class 380, subclass 4.</p> <p data-bbox="618 1640 1382 1703">Group III. . . . drawn to a secure method, classified in Class 380, subclass 3.</p> <p data-bbox="618 1745 1333 1808">Group IV. . . . drawn to [a] method of negotiating electronic contracts, classified in Class 364, subclass 401.</p> <p data-bbox="618 1850 1430 1913">Group V. . . . drawn to methods of auditing a resource, classified in Class 364, subclass 406.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p data-bbox="618 363 1370 430"><del>The inventions are distinct each from the other because of the following reasons:</del></p> <p data-bbox="618 468 1446 850">2. Inventions of Groups I-V are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention of Group I has separate utility such as protecting executable code from computer viruses. Invention of Group II has separate utility such as a computer network administration. Invention of Group III has separate utility such as protection of software. Invention of Group IV has separate utility such as a contract bidding procedure. Invention of Group V has separate utility such as auditing pay television. . . .</p> <p data-bbox="618 888 1438 1024">3. Because <del>these inventions are distinct for the reasons given above and have acquired a separate status in the art</del> as shown by their different classification, restriction for examination purposes as indicated is proper.</p> <p data-bbox="618 1062 1438 1199">4. Because <del>these inventions are distinct for the reasons given above and have acquired a separate status in the art because of their recognized divergent subject matter</del>, restriction for examination purposes as indicated is proper.</p> <p data-bbox="521 1236 1414 1335">'193 File History, 9/25/96 Office Action, pp. 2-3 (a complete copy of this document is attached to the Declaration of Douglas K. Derwin In Support of InterTrust's Claim Construction Position).</p> |

|                       | Claim Term / Phrase  | InterTrust Evidence   |                          |        |             |                    |                       |  |                |                          |
|-----------------------|--|---|--------------------------|--------|-------------|--------------------|-----------------------|--|----------------|--------------------------|
| 25.                   | 193.1: "a budget specifying the number of copies which can be made of said digital file" | <p><b><u>Patent Specifications</u></b></p> <p><b>25(A)</b></p> <p>Traveling objects can also be used to facilitate "moving" an object from one electronic appliance 600 to another. A user could move a traveling object, with its incorporated one or more permission records 808 from a desktop computer, for example, to his notebook computer. A traveling object might register its user within itself and thereafter only be useable by that one user. A traveling object might maintain separate budget information, one for the basic distribution budget record, and another for the "active" distribution budget record of the registered user. In this way, the object could be copied and passed to another potential user, and then could be a portable object for that user.</p> <p>'193 patent at 133:39-50.</p> <hr/> <p><b>25(B)</b></p> <p>Meters and budgets are perhaps among the most common data structures in VDE 100. They are used to count and record events, and also to limit events. The data structures for each meter and budget are determined by the content provider or a distributor/redistributor authorized to change the information. Meters and budgets, however, generally have common information stored in a common header format (e.g., user ID, site ID and related identification information).</p> <p>The content provider or distributor/redistributor may specify data structures for each meter and budget UDE. Although these data structures vary depending upon the particular application, some are more common than others. The following table lists some of the more commonly occurring data structures for METER and BUDGET methods:</p> <table><tr><th>Field type</th><th>Format</th><th>Typical Use</th><th>Description or Use</th></tr><tr><td>Ascending Use Counter</td><td>byte, short, long, or unsigned versions of the</td><td>Meter / Budget</td><td>Ascending count of uses.</td></tr></table> | Field type               | Format | Typical Use | Description or Use | Ascending Use Counter | byte, short, long, or unsigned versions of the | Meter / Budget | Ascending count of uses. |
| Field type            | Format   | Typical Use   | Description or Use       |        |             |                    |                       |  |                |                          |
| Ascending Use Counter | byte, short, long, or unsigned versions of the   | Meter / Budget  | Ascending count of uses. |        |             |                    |                       |  |                |                          |

|  | Claim Term / Phrase | InterTrust Evidence   |  |                |   |
|--|---------------------|---|--|----------------|---|
|  |                     | Descending Use Counter  | same widths  | Budget         | Descending count of permitted use, eg, remaining budget.                              |
|  |                     |   | byte, short, long, or unsigned versions of the same widths   |                |   |
|  |                     | Counter / Limit   | 2, 4 or 8 byte integer split into two related bytes or words | Meter / Budget | usage limits since a specific time, generally used in compound meter data structures. |
|  |                     |   | Array bytes  |                |   |
|  |                     | Bitmap  | Array of bytes   | Meter / Budget | Bit indicator of use or ownership.  |
|  |                     | Wide bitmap   | time_t   | Meter / Budget | Indicator of use or ownership that may age with time.                                 |
|  |                     | Last Use Date   | time_t   | Meter / Budget | Date of last use.   |
|  |                     | Start Date  | time_t   | Budget         | Date of first allowable use.  |
|  |                     | Expiration Date   | time_t   | Meter / Budget | Expiration Date.  |
|  |                     | Last Audit Date   | time_t   | Meter / Budget | Date of last audit.   |
|  |                     | Next Audit Date   | time_t   | Meter / Budget | Date of next required audit.  |
|  |                     | Auditor   | VDE ID   | Meter / Budget | VDE ID of authorized auditor.   |
|  |                     | <p>The information in the table above is not complete or comprehensive, but rather is intended to show some examples of types of information that may be stored in meter and budget related data structures. The actual structure of particular meters and budgets is determined by one or more DTDs 1108 associated with</p> |  |                |   |



|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>the load modules 1100 that create and manipulate the data structure. A list of data types permitted by the DTD interpreter 590 in VDE 100 is extensible by properly authorized parties.</p> <p>'193 patent at 143:38-144:31.</p> <hr/> <p><b>25(C)</b></p> <p>During the same or different communications exchange, the same or different clearinghouse may handle the end user's request for additional budget and/or permission pertaining to VDE object 300. For example, the end user's electronic appliance 600 may (e.g., in response to a user input request to access a particular VDE object 300) send an administrative object to the clearinghouse requesting budgets and/or other permissions allowing access (Block 1164). As mentioned above, such requests may be transmitted in the form of one or more administrative objects, such as, for example, a single administrative object having multiple events associated with multiple requested budgets and/or other permissions for the same or different VDE objects 300. The clearinghouse may upon receipt of such a request, check the end user's credit, financial records, business agreements and/or audit histories to determine whether the requested budgets and/or permissions should be given. The clearinghouse may, based on this analysis, send one or more responsive administrative objects which cause the end user's electronic appliance 600 to update its secure database in response (Block 1166, 1168). This updating might, for example, comprise replacing an expired PERC 808 with a fresh one, modifying a PERC to provide additional (or lesser) rights, etc. Steps 1164-1168 may be repeated multiple times in the same or different communications session to provide further updates to the end user's secure database 610.</p> <p>'193 patent at 162:39-65.</p> <hr/> <p><b>25(D)</b></p> <p>In the example shown in Figure 41d, a distributor at a VDE distributor node (106) might request budget from a content creator at another node (102). This request may be made in the context of a secure VDE communication or it may be passed in an "out-of-channel" communication (e.g. a telephone call or letter). The creator 102 may decide to grant budget to the distributor 106 and</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>processes a distribute event (1452) in BUDGET method 1510 at VDE node 102). A result of processing the distribute event within the BUDGET method might be a secure communication (1454) between VDE nodes 102 and 106 by which a budget granting use and redistribute rights to the distributor 106 may be transferred from the creator 102 to the distributor. The distributor's VDE node 106 may respond to the receipt of the budget information by processing the communication using the reply process 1475B of the BUDGET method 1510. The reply event processing 1475B might, for example, install a budget and PERC 808 within the distributor's VDE 106 node to permit the distributor to access content or processes for which access is control at least in part by the budget and/or PERC. At some point, the distributor 106 may also desire to use the content to which she has been granted rights to access.</p> <p>After registering to use the content object, the user 112 would be required to utilize an array of "use" processes 1476C to, for example, open, read, write, and/or close the content object as part of the use process.</p> <p>Once the distributor 106 has used some or all of her budget, she may desire to obtain additional budget. The distributor 106 might then initiate a process using the BUDGET method request process (1480B). Request process 1480B might initiate a communication (1482AB) with the content creator VDE node 102 requesting more budget and perhaps providing details of the use activity to date (e.g., audit trails). The content creator 102 processes the 'get more budget' request event 1482AB using the response process (1484A) within the creator's BUDGET method 1510A. Response process 1484A might, for example, make a determination if the use information indicates proper use of the content, and/or if the distributor is credit worthy for more budget. The BUDGET method response process 1484A might also initiate a financial transaction to transfer funds from the distributor to pay for said use, or use the distribute process 1472A to distribute budget to the distributor 106. A response to the distributor 106 granting more budget (or denying more budget) might be sent immediately as a response to the request communication 1482AB, or it might be sent at a later time as part of a separate communication. The response communication, upon being received at the distributor's VDE node 106, might be processed using the reply process 1475B within the distributor's copy of the BUDGET method 1510B. The reply process 1475B might then process the additional budget in the same manner as described above.</p> <p>The chain of handling and control may, in addition to posting</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="613 327 1453 747">budget information, also pass control information that governs the manner in which said budget may be utilized. For example, the control information specified in the above example may also contain control information describing the process and limits that apply to the distributor's redistribution of the right to use the creator's content object. Thus, when the distributor responds to a budget request from a user (a communication between a user at VDE node 112 to the distributor at VDE node 106 similar in nature to the one described above between VDE nodes 106 and 102) using the distribute process 1472B within the distributor's copy of the BUDGET method 1510B, a distribution and request/response/reply process similar to the one described above might be initiated.</p> <p data-bbox="521 783 883 816">'193 patent at 172:61-174:29.</p> <hr data-bbox="516 856 1453 861"/> <p data-bbox="521 898 594 932"><b>25(E)</b></p> <p data-bbox="613 968 1305 1001">Transportability of VDE Installations Between PPEs 650</p> <p data-bbox="613 1037 1453 1241">In a preferred embodiment, VDE objects 300 and other secure information may, if appropriate, be transported from one PPE 650 to another securely using the various keys outlined above. VDE 100 uses redistribution of VDE administrative information to exchange ownership of VDE object 300, and to allow the portability of objects between electronic appliances 600.</p> <p data-bbox="613 1276 1453 1690">The permissions record 808 of VDE objects 300 contains rights information that may be used to determine whether an object can be redistributed in whole, in part, or at all. If a VDE object 300 can be redistributed, then electronic appliance 600 normally must have a "budget" and/or other permissioning that allows it to redistribute the object. For example, an electronic appliance 600 authorized to redistribute an object may create an administrative object containing a budget or rights less than or equal to the budget or rights that it owns. Some administrative objects may be sent to other PPEs 650. A PPE 650 that receives one of the administrative objects may have the ability to use at least a portion of the budgets, or rights, to related objects.</p> <p data-bbox="521 1726 824 1759">'193 patent at 220:20-40.</p> <hr data-bbox="516 1799 1453 1803"/> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="516 331 587 363"><b>25(F)</b></p> <p data-bbox="607 401 1430 573">In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).</p> <p data-bbox="516 611 808 642">'193 patent at 48:29-35.</p> <hr data-bbox="516 680 1446 688"/> <p data-bbox="516 726 587 758"><b>25(G)</b></p> <p data-bbox="607 810 1438 1356">... plural, different control models regulating the use and/or auditing of either the same specific copy of electronic information content and/or differently regulating different copies (occurrences) of the same electronic information content. Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments. This includes supporting variable control information for budgeting and auditing usage as applied to a variety of predefined increments of electronic information, including employing a variety of different budgets and/or metering increments for a given electronic information deliverable for: billing units of measure, credit limit, security budget limit and security content metering increments, and/or market surveying and customer profiling content metering increments.</p> <p data-bbox="516 1394 808 1425">'193 patent at 28:19-37.</p> <hr data-bbox="516 1463 1446 1472"/> <p data-bbox="516 1509 587 1541"><b>25(H)</b></p> <p data-bbox="607 1583 1443 1919">... support the flowing of content control information through different "branches" of content control information handling so as to accommodate, under the present invention's preferred embodiment, diverse controlled distributions of VDE controlled content. This allows different parties to employ the same initial electronic content with differing (perhaps competitive) control strategies. In this instance, a party who first placed control information on content can make certain control assumptions and these assumptions would evolve into more specific and/or extensive control assumptions. These control assumptions can evolve during the branching</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>sequence upon content model participants submitting control information changes, for example, for use in "negotiating" with "in place" content control information. This can result in new or modified content control information and/or it might involve the selection of certain one or more already "in-place" content usage control methods over in-place alternative methods, as well as the submission of relevant control information parameter data. This form of evolution of different control information sets applied to different copies of the same electronic property content and/or appliance results from VDE control information flowing "down" through different branches in an overall pathway of handling and control and being modified differently as it diverges down these different pathway branches.</p> <p>'193 patent at 31:29-56.</p> <hr/> <p><b>25(I)</b></p> <p>... multiple simultaneous control models for the same content property and/or property portion. This allows, for example, for concurrent business activities which are dependent on electronic commercial product content distribution, such as acquiring detailed market survey information and/or supporting advertising, both of which can increase revenue and result in lower content costs to users and greater value to content providers. Such control information and/or overall control models may be applied, as determined or allowed by control information, in differing manners to different participants in a pathway of content, reporting, payment, and/or related control information handling. VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities, and/or to different parties in a content and/or appliance usage model, such that different parties (or classes of VDE users, for example) are subject to differing control information managing their use of electronic information content. For example, differing control models based on the category of a user as a distributor of a VDE controlled content object or an end-user of such content may result in different budgets being applied. Alternatively, for example, a one distributor may have the right to distribute a different array of properties than another distributor (from a common content collection provided, for example, on optical disc). An individual, and/or a class or other grouping of end-users, may have different costs (for example, a student, senior citizen, and/or poor citizen user of content who may</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="618 331 1419 401">be provided with the same or differing discounts) than a "typical" content user.</p> <p data-bbox="527 436 846 470">'93 patent at 30:42-31:7.</p> <hr data-bbox="527 510 1461 514"/> <p data-bbox="527 552 592 585">25(J)</p> <p data-bbox="618 621 1451 1241">Such different application of control information may also result from content control information specifying that a certain party or group of parties shall be subject to content control information that differs from another party or group of parties. For example, content control information for a given piece of content may be stipulated as senior information and therefore not changeable, might be put in place by a content creator and might stipulate that national distributors of a given piece of their content may be permitted to make 100,000 copies per calendar quarter, so long as such copies are provided to bona fide end-users, but may pass only a single copy of such content to a local retailers and the control information limits such a retailer to making no more than 1,000 copies per month for retail sales to end-users. In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).</p> <p data-bbox="527 1276 818 1310">'93 patent at 48:15-35.</p> <hr data-bbox="527 1350 1461 1354"/> <p data-bbox="527 1392 597 1425">25(K)</p> <p data-bbox="618 1461 1451 1906">In this example, as illustrated in Figure 80, user B may receive control information associated with creator A's content from distributor A and/or user/distributor B. In either case, user B may be able to establish their own control information on DA(CA) and/or UDB(UDA(DA(CA))), respectively (if allowed by such control information. The resulting set(s) of control information UB(DA(CA)) and/or UB(UDB(UDA(DA(CA)))) respectively, may represent different control scenarios, each of which may have benefits for user B. As described in connection with an earlier example, user B may have received control information from user/distributor B along a chain of handling including user/distributor A that bases fees on the number of minutes that user B makes use of creator A's content (and requiring user/distributor A</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>to pay fees of \$15 per month per user to distributor A regardless of the amount of usage by user B in a calendar month). This may be more favorable under some circumstances than the fees required by a direct use of control information provided by distributor A, but may also have the disadvantage of an exhausted chain of redistribution and, for example, further usage information reporting requirements included in UDB(UDA(DA(CA))). If the two sets of control information DA(CA) and UDB(UDA(DA(CA))) permit (e.g. do not require exclusivity enforced, for example, by using a registration interval in an object registry used by a secure subsystem of user B's VDE installation to prevent deregistration and reregistration of different sets of control information related to a certain container (or registration of plural copies of the same content having different control information and/or being supplied by different content providers) within a particular interval of time as an aspect of an extended agreement for a chain of handling and control reflected in DA(CA) and/or UDB(UDA(DA(CA))) , user B may have both sets of control information registered and may make use of the set that they find preferable under a given usage scenario.</p> <p>'193 patent at 306:30-65.</p> <hr/> <p><b>25(L)</b></p> <p>For example, user/distributor A may receive control information CB that includes a requirement that user/distributor A pay creator B for content decrypted by user/distributor A (and any participant receiving distributed and/or redistributed control information from user/distributor A) at the rate of \$0.50 per kilobyte. As indicated above, user/distributor A also may receive control information associated with creator B's VDE content container from distributor A. In this example, user/distributor A may have a choice between paying a "rental" fee through a chain of handling passing through distributor A, and a fee based on the quantity of decryption through a chain of handling direct to creator B. In this case, user/distributor A may have the ability to choose to use either or both of CB and DA(CB).</p> <p>'193 patent at 308:29-42.</p> <hr/> <p><b>25(M)</b></p> <p>As illustrated in Figure 81, in this example, user B may receive</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>control information associated with creator B's VDE content container from six different sources: CB directly from creator B, DA(CB) from distributor A, UDB(UDA(DA(CB))) and/or UDB(UDA(CB)) from user/distributor B, DC(CB) from distributor C, and/or DB(DC(CB)) from distributor B. This represents six chains of handling through which user B may enter into extended agreements with other participants in this example. Two of these chains pass through user/distributor B. Based on a VDE negotiation between user/distributor B and user B, an extended agreement may be reached (if permitted by control information governing both parties) that reflects the conditions under which user B may use one or both sets of control information. In this example, two chains of handling and control may "converge" at user/distributor B, and then pass to user B (and if control information permits, later diverge once again based on distribution and/or redistribution by user B).</p> <p>'193 patent at 308:48-65.</p> <hr/> <p>25(N)</p> <p>User B may, in this example, receive a VDE content container from distributor C that is comprised of VDE objects created by creator B, creator C, and creator D. In addition, user B may receive a VDE content container from distributor B that contains the same content created by creator B, creator C, and creator D in addition to one or more extracted/embedded portions of content created by creator E. User B may base decisions concerning which of such containers they choose to use (including which embedded containers she may wish to use), and under which circumstances, based on, for example, the character of such extracted/embedded portions (e.g. multimedia presentations illustrating potential areas of interest in the remainder of the content, commentary explaining and/or expositing other elements of content, related works, improved application software delivered as an element of content, etc.); the quality, utility, and/or price (or other attributes of control information) of such portions; and other considerations which distinguish the containers and/or content control information received, in this example, from distributor B and distributor C.</p> <p>'193 patent at 312:11-31.</p> <hr/> |



|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="527 325 609 357"><b>25(O)</b></p> <p data-bbox="617 394 1429 535">As with standard VDE objects 300, a user may be required to contact a clearinghouse service to acquire additional budgets if the user wishes to continue to use the traveling object after the exhaustion of an available budget(s) . . . .</p> <p data-bbox="527 573 836 604">'93 patent at 131:10-13.</p> <hr data-bbox="527 640 1453 651"/> <p data-bbox="527 682 609 714"><b>25(P)</b></p> <p data-bbox="617 751 1453 1585">Once the distributor 106 has used some or all of her budget, she may desire to obtain additional budget. The distributor 106 might then initiate a process using the BUDGET method request process (1480B). Request process 1480B might initiate a communication (1482AB) with the content creator VDE node 102 requesting more budget and perhaps providing details of the use activity to date (e.g., audit trails). The content creator 102 processes the 'get more budget' request event 1482AB using the response process (1484A) within the creator's BUDGET method 1510A. Response process 1484A might, for example, make a determination if the use information indicates proper use of the content, and/or if the distributor is credit worthy for more budget. The BUDGET method response process 1484A might also initiate a financial transaction to transfer funds from the distributor to pay for said use, or use the distribute process 1472A to distribute budget to the distributor 106. A response to the distributor 106 granting more budget (or denying more budget) might be sent immediately, as a response to the request communication 1482AB, or it might be sent at a later time as part of a separate communication. The response communication, upon being received at the distributor's VDE node 106, might be processed using the reply process 1475B within the distributor's copy of the BUDGET method 1510B. The reply process 1475B might then process the additional budget in the same manner as described above.</p> <p data-bbox="527 1623 885 1654">'93 patent at 173:21-174:14.</p> <hr data-bbox="527 1690 1453 1701"/> <p data-bbox="527 1732 609 1764"><b>25(Q)</b></p> <p data-bbox="617 1801 1453 1904">During the same or different communications exchange, the same or different clearinghouse may handle the end user's request for additional budget and/or permission pertaining to VDE object 300.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>For example, the end user's electronic appliance 600 may (e.g., in response to a user input request to access a particular VDE object 300) send an administrative object to the clearinghouse requesting budgets and/or other permissions allowing access (Block 1164). As mentioned above, such requests may be transmitted in the form of one or more administrative objects, such as, for example, a single administrative object having multiple "events" associated with multiple requested budgets and/or other permissions for the same or different VDE objects 300. The clearinghouse may upon receipt of such a request, check the end user's credit, financial records, business agreements and/or audit histories to determine whether the requested budgets and/or permissions should be given. The clearinghouse may, based on this analysis, send one or more responsive administrative objects which cause the end user's electronic appliance 600 to update its secure database in response (Block 1166, 1168). This updating might, for example, comprise replacing an expired PERC 808 with a fresh one, modifying a PERC to provide additional (or lesser) rights, etc. Steps 1164-1168 may be repeated multiple times in the same or different communications session to provide further updates to the end user's secure database 610.</p> <p>'193 patent at 162:39-65.</p> |

|     | Claim Term / Phrase  | InterTrust Evidence   |
|-----|--|---|
| 26. | 193.1:<br>"controlling the copies made of said digital file" | <p data-bbox="526 323 797 359"><u>Patent Specifications</u></p> <p data-bbox="526 394 602 430">26(A)</p> <p data-bbox="613 478 1446 1031">... plural, different control models regulating the use and/or auditing of either the same specific copy of electronic information content and/or differently regulating different copies (occurrences) of the same electronic information content. Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments. This includes supporting variable control information for budgeting and auditing usage as applied to a variety of predefined increments of electronic information, including employing a variety of different budgets and/or metering increments for a given electronic information deliverable for: billing units of measure, credit limit, security budget limit and security content metering increments, and/or market surveying and customer profiling content metering increments.</p> <p data-bbox="526 1073 813 1108">'193 patent at 28:19-37.</p> <hr data-bbox="526 1142 1446 1150"/> <p data-bbox="526 1184 602 1220">26(B)</p> <p data-bbox="613 1255 1446 1906">... support the flowing of content control information through different "branches" of content control information handling so as to accommodate, under the present invention's preferred embodiment, diverse controlled distributions of VDE controlled content. This allows different parties to employ the same initial electronic content with differing (perhaps competitive) control strategies. In this instance, a party who first placed control information on content can make certain control assumptions and these assumptions would evolve into more specific and/or extensive control assumptions. These control assumptions can evolve during the branching sequence upon content model participants submitting control information changes, for example, for use in "negotiating" with "in place" content control information. This can result in new or modified content control information and/or it might involve the selection of certain one or more already "in-place" content usage control methods over in-place alternative methods, as well as the submission of relevant control information parameter data. This form of evolution of different control information sets applied to different copies of the same electronic property content and/or</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>appliance results from VDE control information flowing "down" through different branches in an overall pathway of handling and control and being modified differently as it diverges down these different pathway branches.</p> <p>'193 patent at 31:29-56.</p> <hr/> <p><b>26(C)</b></p> <p>... multiple simultaneous control models for the same content property and/or property portion. This allows, for example, for concurrent business activities which are dependent on electronic commercial product content distribution, such as acquiring detailed market survey information and/or supporting advertising, both of which can increase revenue and result in lower content costs to users and greater value to content providers. Such control information and/or overall control models may be applied, as determined or allowed by control information, in differing manners to different participants in a pathway of content, reporting, payment, and/or related control information handling. VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities, and/or to different parties in a content and/or appliance usage model, such that different parties (or classes of VDE users, for example) are subject to differing control information managing their use of electronic information content. For example, differing control models based on the category of a user as a distributor of a VDE controlled content object or an end-user of such content may result in different budgets being applied. Alternatively, for example, a one distributor may have the right to distribute a different array of properties than another distributor (from a common content collection provided, for example, on optical disc). An individual, and/or a class or other grouping of end-users, may have different costs (for example, a student, senior citizen, and/or poor citizen user of content who may be provided with the same or differing discounts) than a typical content user.</p> <p>'193 patent at 30:42-31:7.</p> <hr/> <p><b>26(D)</b></p> <p>Such different application of control information may also result from content control information specifying that a certain party or</p> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p>group of parties shall be subject to content control information that differs from another party or group of parties. For example, content control information for a given piece of content may be stipulated as senior information and therefore not changeable, might be put in place by a content creator and might stipulate that national distributors of a given piece of their content may be permitted to make 100,000 copies per calendar quarter, so long as such copies are provided to bona fide end-users, but may pass only a single copy of such content to a local retailers and the control information limits such a retailer to making no more than 1,000 copies per month for retail sales to end-users. In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).</p> <p>'193 patent at 48:15-35.</p> <hr/> <p>26(E)</p> <p>In this example, as illustrated in Figure 80, user B may receive control information associated with creator A's content from distributor A and/or user/distributor B. In either case, user B may be able to establish their own control information on DA(CA) and/or UDB(UDA(DA(CA))), respectively (if allowed by such control information. The resulting set(s) of control information UB(DA(CA)) and/or UB(UDB(UDA(DA(CA)))) respectively, may represent different control scenarios, each of which may have benefits for user B. As described in connection with an earlier example, user B may have received control information from user/distributor B along a chain of handling including user/distributor A that bases fees on the number of minutes that user B makes use of creator A's content (and requiring user/distributor A to pay fees of \$15 per month per user to distributor A regardless of the amount of usage by user B in a calendar month). This may be more favorable under some circumstances than the fees required by a direct use of control information provided by distributor A, but may also have the disadvantage of an exhausted chain of redistribution and, for example, further usage information reporting requirements included in UDB(UDA(DA(CA))). If the two sets of control information DA(CA) and UDB(UDA(DA(CA))) permit (e.g. do not require exclusivity enforced, for example, by using a registration interval in an object registry used by a secure subsystem of user B's VDE installation to prevent deregistration and</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p>built to execute only in SPE 503 secure space for government classes of security, or in an HPE 655 for commercial applications. As described above, the load module public header 802 may contain an "execution space code" field that indicates where the load module 1100 needs to execute.</p> <p>'193 patent at 140:15-46.</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p>reregistration of different sets of control information related to a certain container (or registration of plural copies of the same content having different control information and/or being supplied by different content providers) within a particular interval of time as an aspect of an extended agreement for a chain of handling and control reflected in DA(CA) and/or UDB(UDA(DA(CA))) , user B may have both sets of control information registered and may make use of the set that they find preferable under a given usage scenario.</p> <p>'193 patent at 306:30-65.</p> <hr/> <p><b>26(F)</b></p> <p>For example, user/distributor A may receive control information CB that includes a requirement that user/distributor A pay creator B for content decrypted by user/distributor A (and any participant receiving distributed and/or redistributed control information from user/distributor A) at the rate of \$0.50 per kilobyte. As indicated above, user/distributor A also may receive control information associated with creator B's VDE content container from distributor A. In this example, user/distributor A may have a choice between paying a "rental" fee through a chain of handling passing through distributor A, and a fee based on the quantity of decryption through a chain of handling direct to creator B. In this case, user/distributor A may have the ability to choose to use either or both of CB and DA(CB).</p> <p>'193 patent at 308:29-42.</p> <hr/> <p><b>26(G)</b></p> <p>As illustrated in Figure 81, in this example, user B may receive control information associated with creator B's VDE content container from six different sources: CB directly from creator B, DA(CB) from distributor A, UDB(UDA(DA(CB))) and/or UDB(UDA(CB)) from user/distributor B, DC(CB) from distributor C, and/or DB(DC(CB)) from distributor B. This represents six chains of handling through which user B may enter into extended agreements with other participants in this example. Two of these chains pass through user/distributor B. Based on a VDE negotiation between user/distributor B and user B, an extended agreement may be reached (if permitted by control information governing both parties) that reflects the conditions under which user B may use one</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>or both sets of control information. In this example, two chains of handling and control may "converge" at user/distributor B, and then pass to user B (and if control information permits, later diverge once again based on distribution and/or redistribution by user B).</p> <p>'193 patent at 308:48-65.</p> <hr/> <p><b>26(H)</b></p> <p>User B may, in this example, receive a VDE content container from distributor C that is comprised of VDE objects created by creator B, creator C, and creator D. In addition, user B may receive a VDE content container from distributor B that contains the same content created by creator B, creator C, and creator D in addition to one or more extracted/embedded portions of content created by creator E. User B may base decisions concerning which of such containers they choose to use (including which embedded containers she may wish to use), and under which circumstances, based on, for example, the character of such extracted/embedded portions (e.g. multimedia presentations illustrating potential areas of interest in the remainder of the content, commentary explaining and/or expositing other elements of content, related works, improved application software delivered as an element of content, etc.); the quality, utility, and/or price (or other attributes of control information) of such portions; and other considerations which distinguish the containers and/or content control information received, in this example, from distributor B and distributor C.</p> <p>'193 patent at 312:11-31.</p> |



|     | Claim Term / Phrase  | InterTrust Evidence   |
|-----|--|---|
| 27. | 721.1: "digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class" | <p><b><u>Patent Specifications</u></b></p> <p><b>27(A)</b></p> <p>In accordance with one aspect provided by the present invention, one or more trusted verifying authorities validate load modules or other executables by analyzing and/or testing them. A verifying authority digitally "signs" and "certifies" those load modules or other executables it has verified (using a public key based digital signature and/or certificate based thereon, for example).</p> <p>Protected execution spaces such as protected processing environments can be programmed or otherwise conditioned to accept only those load modules or other executables bearing a digital signature/certificate of an accredited (or particular) verifying authority.</p> <p>'721 patent at 4:61-5:5.</p> <hr/> <p><b>27(B)</b></p> <p>A hierarchy of assurance levels may be provided for different protected processing environment security levels. Load modules or other executables can be provided with digital signatures associated with particular assurance levels. Appliances assigned to particular assurance levels can protect themselves from executing load modules or other executables associated with different assurance levels. Different digital signatures and/or certificates may be used to distinguish between load modules or other executables intended for different assurance levels. This strict assurance level hierarchy provides a framework to help ensure that a more trusted environment can protect itself from load modules or other executables exposed to environments with different work factors (e.g., less trusted or tamper resistant environments). This can be used to provide a high degree of security compartmentalization that helps protect the remainder of the system should parts of the system become compromised.</p> <p>For example, protected processing environments or other secure execution spaces that are more impervious to tampering (such as those providing a higher degree of physical security) may use an assurance level that isolates it from protected processing environments or other secure execution spaces that are relatively more susceptible to tampering (such as those constructed solely by software executing on a general purpose digital computer in a non-</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p>secure location).</p> <p>A verifying authority may digitally sign load modules or other executables with a digital signature that indicates or implies assurance level. A verifying authority can use digital signature techniques to distinguish between assurance levels. As one example, each different digital signature may be encrypted using a different verification key and/or fundamentally different encryption, one-way hash and/or other techniques. A protected processing environment or other secure execution space protects itself by executing only those load modules or other executables that have been digitally signed for its corresponding assurance level.</p> <p>The present invention may use a verifying authority and the digital signatures it provides to compartmentalize the different electronic appliances depending on their level of security (e.g., work factor or relative tamper resistance). In particular, a verifying authority and the digital signatures it provides isolate appliances with significantly different work factors — preventing the security of high work factor appliances from collapsing into the security of low work factor appliances due to free exchange of load modules or other executables.</p> <p>'721 patent at 6:16-62.</p> <hr/> <p>27(C)</p> <p>Figures 11A-11C show how a verifying authority can use different digital signatures to designate the same or different load modules as being appropriate for execution by different assurance level electronic appliances.</p> <p>Figures 12, 13 and 13A show how assurance level digital signatures can be used to isolate electronic appliances or appliance types based on work factor and/or tamper resistance to reduce overall security risks.</p> <p>'721 patent at 7:66-8:6.</p> <hr/> |

|  | Claim Term / Phrase | InterTrust Evidence  |
|--|---------------------|--|
|  |                     | <p data-bbox="521 331 597 369"><b>27(D)</b></p> <p data-bbox="613 405 834 436">Assurance Levels</p> <p data-bbox="613 472 1446 682">Verifying authority 100 can use different digital signing techniques to provide different "assurance levels" for different kinds of electronic appliances 61 having different "work factors" or levels of tamper resistance. Figures 10A-10C show an example assurance level hierarchy providing three different assurance levels for different electronic appliance types:</p> <p data-bbox="613 718 1446 928">Assurance level I might be used for an electronic appliance(s) 61 whose protected processing environment 108 is based on software techniques that may be somewhat resistant to tampering. An example of an assurance level I electronic appliance 61A might be a general purpose personal computer that executes software to create protected processing environment 108.</p> <p data-bbox="613 963 1446 1306">An assurance level II electronic appliance 61B may provide a protected processing environment 108 based on a hybrid of software security techniques and hardware-based security techniques. An example of an assurance level II electronic appliance 61B might be a general purpose personal computer equipped with a hardware integrated circuit secure processing unit ("SPU") that performs some secure processing outside of the SPU (see Ginter et al. patent disclosure Figure 10 and associated text). Such a hybrid arrangement might be relatively more resistant to tampering than a software-only implementation.</p> <p data-bbox="613 1341 1446 1621">The assurance level III appliance 61C shown is a general purpose personal computer equipped with a hardware-based secure processing unit 132 providing and completely containing protected processing environment 108 (see Ginter et al. Figures 6 and 9 for example). A silicon-based special purpose integrated circuit security chip is relatively more tamper-resistant than implementations relying on software techniques for some or all of their tamper-resistance.</p> <p data-bbox="613 1656 1446 1866">In this example, verifying authority 100 digitally signs load modules 54 using different digital signature techniques (for example, different "private" keys 122) based on assurance level. The digital signatures 106 applied by verifying authority 100 thus securely encode the same (or different) load module 54 for use by appropriate corresponding assurance level electronic appliances 61.</p> <p data-bbox="613 1902 1386 1934">Assurance level in this example may be assigned to a particular</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>protected processing environment 108 at initialization (e.g., at the factory in the case of hardware-based secure processing units). Assigning assurance level at initialization time facilitates the use of key management (e.g., secure key exchange protocols) to enforce isolation based on assurance level. For example, since establishment of assurance level is done at initialization time, rather than in the field in this example, the key exchange mechanism can be used to provide new keys (assuming an assurance level has been established correctly).</p> <p>'721 patent at 16:37-17:23.</p> <hr/> <p><b>27(E)</b></p> <p>In one example, verifying authority 100 may digitally sign identical copies of load module 54 for use by different classes or "assurance levels" of electronic appliances 61. If the sharing of a load module 54 between different electronic appliances is regarded as an open communications channel between the protected processing environments 108 of the two appliances, it becomes apparent that there is a high degree of risk in permitting such sharing to occur. In particular, the extra security assurances and precautions of the more trusted environment are collapsed into the those of the less trusted environment because an attacker who compromises a load module within a less trusted environment is then be able to launch the same load module to attack the more trusted environment. Hence, although compartmentalization based on encryption and key management can be used to restrict certain kinds of load modules 54 to execute only on certain types of electronic appliances 61, a significant application in this context is to compartmentalize the different types of electronic appliances and thereby allow an electronic appliance to protect itself against load modules 54 of different assurance levels.</p> <p>'721 patent at 18:19-38.</p> <hr/> <p><b>27(F)</b></p> <p>In accordance with this feature of the invention, verifying authority 100 supports all of these various categories of digital signatures, and system 50 uses key management to distribute the appropriate verification keys to different assurance level devices. For example, verifying authority 100 may digitally sign a particular load module</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="618 327 1451 537">54 such that only hardware-only based server(s) 402(3) at assurance level XI may authenticate it. This compartmentalization prevents any load module executable on hardware-only servers 402(3) from executing on any other assurance level appliance (for example software-only protected processing environment based support service 404(1)).</p> <p data-bbox="618 575 1451 919">To simplify key management and distribution, execution environments having significantly similar work factors can be classified in the same assurance level. Figure 13 shows one example hierarchical assurance level arrangement. In this example, less secure “software only” protected processing environment 108 devices are categorized as assurance level I, somewhat more secure “software and hardware hybrid” protected processing environment appliances are categorized as assurance level II, and more trusted “hardware only” protected processing environment devices are categorized as assurance level III.</p> <p data-bbox="526 957 821 989">‘721 patent at 19:11-32.</p> <hr data-bbox="526 1024 1451 1035"/> <p data-bbox="526 1073 602 1104"><b>27(G)</b></p> <p data-bbox="618 1142 1451 1276">A load module or other executable may be certified for multiple assurance levels. Different digital signatures may be used to certify the same load module or other executable for different respective assurance levels.</p> <p data-bbox="526 1314 789 1346">‘721 patent at 20:1-4.</p> |

|     | Claim Term / Phrase  | InterTrust Evidence   |
|-----|--|---|
| 28. | 891.1: "securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item" | <p><u>Patent Specifications</u></p> <p>28(A)</p> <p>The embedding processes for all VDE embedded content containers normally involves securely identifying the appropriate content control information for the embedded content. For example, VDE content control information for a VDE installation and/or a VDE content container may securely, and transparently to an embedder (user), apply the same content control information to edited (such as modified or additional) container content as is applied to one or more portions (including all, for example) of previously "in place" content of said container and/or <u>securely apply control information</u> generated through a VDE control information negotiation between control sets, and/or it may apply control information previously applied to said content. Application of control information may occur regardless of whether the edited content is in a parent or embedded container. <u>This same capability of securely applying content control information</u> (which may be automatically and/or transparently applied), may also be employed with content that is embedded into a VDE container through extracting and embedding content, or through the moving, or copying and embedding, of VDE container objects. Application of content control information normally occurs securely within one or more VDE secure sub-system PPEs 650. This process may employ a VDE template that enables a user, through easy to use GUI user interface tools, to specify VDE content control information for certain or all embedded content, and which may include menu driven, user selectable and/or definable options, such as picking amongst alternative control methods (e.g. between different forms of metering) which may be represented by different icons picturing (symbolizing) different control functions and apply such functions to an increment of VDE secured content, such as an embedded object listed on an object directory display.</p> <p>'193 patent at 299:19-51.</p> <hr/> <p>28(B)</p> <p>Embedded content (and/or content objects) may have been contributed by different parties and may be integrated into a VDE container through a VDE content and content control information integration process securely managed through the use of one or more secure VDE subsystems. This process may, for example, involve one or more of:</p> |

|  | Claim Term / Phrase | InterTrust Evidence   |
|--|---------------------|---|
|  |                     | <p data-bbox="613 363 1451 919">(1.) <u>securely applying instructions controlling the embedding and/or use of said submitted content</u>, wherein said instructions were securely put in place, at least in part, by a content provider and/or user of said VDE container. For example, said user and/or provider may interact with one or more user interfaces offering a selection of content embedding and/or control options (e.g. in the form of a VDE template). Such options may include which, and/or whether, one or more controls should be applied to one or more portions of said content and/or the entry of content control parameter data (such a time period before which said content may not be used, cost of use of content, and/or pricing discount control parameters such as software program suite sale discounting). Once required and/or optional content control information is established by a provider and/or user, it may function as content control information which may be, in part or in full, applied automatically to certain, or all, content which is embedded in a VDE content container.</p> <p data-bbox="521 957 813 989">'193 patent at 300:6-30.</p> <hr/> <p data-bbox="521 1068 591 1100">28(C)</p> <p data-bbox="613 1138 1442 1308">Users of VDE may include content <u>creators who apply content usage, usage reporting, and/or usage payment related control information to electronic content</u> and/or appliances for users such as end-user organizations, individuals, and content and/or appliance distributors.</p> <p data-bbox="521 1346 797 1377">'193 patent at 9:40-45.</p> <hr/> <p data-bbox="521 1457 591 1488">28(D)</p> <p data-bbox="613 1526 1451 1770">For example, in a VDE aware word processor application, a user may be able to "print" a document into a VDE content container object, <u>applying specific control information by selecting from amongst a series of different menu templates for different purposes</u> (for example, a confidential memo template for internal organization purposes may restrict the ability to "keep," that is to make an electronic copy of the memo).</p> <p data-bbox="521 1808 808 1839">'193 patent at 26:59-67.</p> |

|  | Claim Term /<br>Phrase | InterTrust Evidence   |
|--|------------------------|---|
|  |                        | <p data-bbox="516 331 592 363"><b>28(E)</b></p> <p data-bbox="609 401 1446 716">VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities, and/or to different parties in a content and/or appliance usage model, such that different parties (or classes of VDE users, for example) are subject to differing control information managing their use of electronic information content. For example, differing control models based on the category of a user as a distributor of a VDE controlled content object or an end-user of such content may result in different budgets being applied.</p> <p data-bbox="516 751 808 783">'193 patent at 30:55-65.</p> <hr data-bbox="516 821 1446 825"/> <p data-bbox="516 863 592 894"><b>28(F)</b></p> <p data-bbox="609 932 1382 999">Keys and tags may be securely generated within SPE 503 (HPE 655) in the preferred embodiment.</p> <p data-bbox="516 1037 824 1068">'193 patent at 120:15-16.</p> <hr data-bbox="516 1106 1446 1110"/> <p data-bbox="516 1148 592 1180"><b>28(G)</b></p> <p data-bbox="609 1218 1446 1493">Frequently, for a VDE application for a given content model (such as distribution of entertainment on CD-ROM, content delivery from an Internet repository, or electronic catalog shopping and advertising, or some combination of the above) participants would be able to securely select from amongst available, alternative control methods and apply related parameter data wherein such selection of control method and/or submission of data would constitute their "contribution" of control information.</p> <p data-bbox="516 1530 829 1562">'193 patent at 18:60-19:1.</p> <hr data-bbox="516 1600 1446 1604"/> <p data-bbox="516 1642 592 1673"><b>28(H)</b></p> <p data-bbox="609 1711 1414 1850">ROS 602 assembles these elements together into an executable component assembly 690 prior to loading and executing the component assembly (e.g., in a secure operating environment such as SPE 503 and/or HPE 655).</p> <p data-bbox="516 1887 797 1919">'193 patent at 83:44-48</p> |



|     | Claim Term / Phrase   | InterTrust Evidence   |
|-----|---|---|
| 29. | 900.155: "derives information from one or more aspects of said host processing environment" | <p><b><u>Patent Specifications</u></b></p> <p><b>29(A)</b></p> <p>Correspondence Between Installed Software and Appliance "Signature". Another technique that may be used during the installation routine 3470 is to customize the operational materials 3472 by embedding a "machine signature" into the operational materials to establish a correspondence between the installed software on a particular electronic appliance 600 (Figure 69C, block 3470(7)). This technique prevents a software-based PPE 650 from being transferred from one electronic appliance 600 to another (except through the use of the appropriate secure, verified backup mechanism).</p> <p>For electronic appliances 600 where it is feasible to do so, the installation procedure 3470 may determine unique information about the electronic appliance 600 (e.g., a "signature" SIG in the sense of a unique value, not necessarily a "digital signature" in the cryptographic sense). Installation routine 3470 embeds the electronic appliance "signature" SIG in the installed operational materials 3472. Upon initialization, the operational materials 3472 validate the embedded signature value against the actual electronic appliance 600 signature SIG, and may refuse to start if the comparison fails.</p> <p>Depending on the configuration of electronic appliance 600, the machine signature may consist, for example, of some combination of</p> <ul style="list-style-type: none"> <li>a hash of the ROM BIOS 658 (see Figure 69G);</li> <li>a hash of a disk defect map 3497a;</li> <li>the Ethernet (or other) network adapter 666 address;</li> <li>information written into an unused disk sector;</li> <li>information stored in a non-volatile CMOS RAM (such as used for hardware configuration data);</li> <li>information stored in non-volatile ("flash") memory (such as used for system or peripheral component "BIOS" programs) and/or</li> <li>hidden unique information placed into the root directory 3497b of the fixed disk drive 668;</li> </ul> |

|  | Claim Term /<br>Phrase | InterTrust Evidence  |
|--|------------------------|--|
|  |                        | <p>Figure 69G shows an example of some of these appliance-specific signatures.</p> <p>'900 patent at 239:4-42.</p> |

|                                  | Claim Term / Phrase  | InterTrust Evidence   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
|----------------------------------|--|---|------------|-------------|-------|------------------------|------------|---|---------|--------------------------------------|-------|--|------------|-------------------------------------|----------------------------------|--|--|--|-------------------------|--|--|--|
| 30.                              | 912.8:<br>“identifying at least one aspect of an execution space required for use and/or execution of the load module”   | <p><b>Patent Specifications</b><br/><b>30(A)</b></p> <p>The following is an example of a possible field layout for load module public header 802:</p> <table><tr><th>Field Type</th><th>Description</th></tr><tr><td>LM ID</td><td>VDE ID of Load Module.</td></tr><tr><td>Creator ID</td><td>Site ID of creator of this load module.</td></tr><tr><td>Type ID</td><td>Constant indicates load module type.</td></tr><tr><td>LM ID</td><td>Unique sequence number for this load module, which uniquely identifies the load module in a sequence of load modules created by an authorized VDE participant.</td></tr><tr><td>Version ID</td><td>Version number of this load module.</td></tr><tr><td>Other classification information</td><td>Class ID<br/>ID to support different load module classes.</td></tr><tr><td></td><td>Type ID<br/>ID to support method type compatible searching.</td></tr><tr><td>Descriptive Information</td><td>Description<br/>Textual description of the load module.</td></tr><tr><td></td><td>Execution space code<br/>Value that describes what execution space (e.g., SPE or HPE) this load module.</td></tr></table> <p>Many load modules 1100 contain code that executes in an SPE 503. Some load modules 1100 contain code that executes in an HPE 655. This allows methods 1000 to execute in whichever environment is appropriate. For example, an INFORMATION method 1000 can be built to execute only in SPE 503 secure space for government classes of security, or in an HPE 655 for commercial applications. As described above, the load module public header 802 may contain an “execution space code” field that indicates where the load module 1100 needs to execute.</p> | Field Type | Description | LM ID | VDE ID of Load Module. | Creator ID | Site ID of creator of this load module. | Type ID | Constant indicates load module type. | LM ID | Unique sequence number for this load module, which uniquely identifies the load module in a sequence of load modules created by an authorized VDE participant. | Version ID | Version number of this load module. | Other classification information | Class ID<br>ID to support different load module classes. |  | Type ID<br>ID to support method type compatible searching. | Descriptive Information | Description<br>Textual description of the load module. |  | Execution space code<br>Value that describes what execution space (e.g., SPE or HPE) this load module. |
| Field Type                       | Description  |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| LM ID                            | VDE ID of Load Module.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| Creator ID                       | Site ID of creator of this load module.  |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| Type ID                          | Constant indicates load module type.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| LM ID                            | Unique sequence number for this load module, which uniquely identifies the load module in a sequence of load modules created by an authorized VDE participant. |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| Version ID                       | Version number of this load module.  |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| Other classification information | Class ID<br>ID to support different load module classes.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
|                                  | Type ID<br>ID to support method type compatible searching.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
| Descriptive Information          | Description<br>Textual description of the load module.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
|                                  | Execution space code<br>Value that describes what execution space (e.g., SPE or HPE) this load module.   |   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |
|                                  |  | ‘193 patent at 140:15-46.   |            |             |       |                        |            |   |         |                                      |       |  |            |                                     |                                  |  |  |  |                         |  |  |  |

**EXHIBIT D**  
**PLR 4-3(b) – Microsoft’s Listing of Intrinsic and Extrinsic Evidence**

Set forth below are references to the “intrinsic” and “extrinsic” evidence on which Microsoft may rely to support its claim construction for the 30 designated “Mini-Markman” terms and phrases. Each claim phrase incorporates the intrinsic and extrinsic support of the individual terms within it.

For ease of reference, the full titles of various intrinsic and extrinsic evidence sources are abbreviated. A key to the abbreviations is contained in Appendix 1, located at the last page of this Exhibit.

|    | Claim<br>Term/Phrase                              | Evidence Supporting MS Construction  |
|----|---|--|
| 1. | aspect<br><br>683.2<br>861.58<br>900.155<br>912.8 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “For each site, the manufacturer generates a site ID 2821 and list of site characteristics 2822.” (‘193 209:55-57)</li> <li>2. See also support listed in item #29 (‘900:155)</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Aspect: “The qualification of a descriptor.” (IBM)</li> </ol>   |
| 2. | authentication<br><br>193.15                      | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “A certification key pair may be used as part of a ‘certification’ process for PPEs 650 and VDE electronic appliances 600. This certification process in the preferred embodiment may be used to permit a VDE electronic appliance to present one or more ‘certificates’ authenticating that it (or its key) can be trusted. As described above, this ‘certification’ process may be used by one PPE 650 to ‘certify’ that it is an authentic VDE PPE, it has a certain level of security and capability set (e.g., it is hardware based rather than merely software based), etc.” (‘193 212:66 - 213:15)</li> <li>2. “One of the functions SPU 500 may perform is to validate/authenticate VDE objects 300 and other items. Validation/authentication often involves comparing long data strings to determine whether they compare in a predetermined way.” (‘193 67:56-60)</li> <li>3. “Sender 4052 may select different ways to identify recipients 4056 based on the confidentiality of the document and the level of security the sender is willing to pay for. In one example, sender 4052 might require the recipient’s appliance 600B to require recipient 4056 to prove that he is who he says he is. This secure ‘authentication’ function might be met by, for example, requiring recipient 4056 to input a password, present digital proof of identity...” (‘683 17:20-27)</li> <li>4. “In order to further assure the authenticity of the communication, a secure communications link may be established using a key exchange technique (e.g., Diffie-Hellman) and encryption of the signal between the stations.” (‘683 52:56-60)</li> <li>5. “This ‘channel 0’ ‘open channel’ task may then issue a series of requests to secure database manager 566 to obtain the ‘blueprint’ for constructing one or more component assemblies 690 to be associated with channel 594 (block 1127). In the preferred embodiment, this ‘blueprint’ may comprise a PERC 808 and/or URT 464. In may be obtained by using the ‘Object, User, Right’ parameters passed to the ‘open channel’ routine to ‘chain’ together object registration table</li> </ol> |

|    | Claim<br>Term/Phrase       | Evidence Supporting MS Construction   |
|----|----------------------------|---|
|    |                            | <p>460 records, user/object table 462 records, URT 464 records, and PERC 808 records. This 'open channel' task may preferably place calls to key and tag manager 558 to validate and correlate the tags associated with these various records to ensure that they are authentic and match. The preferred embodiment process then may write appropriate information to channel header 596 (block 1129)." ('193 112:46-61)</p> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Authentication: "1. In computer security, verification of the identity of a user or the user's eligibility to access an object. 2. In computer security, verification that a message has not been altered or corrupted. 3. In computer security, a process used to verify the user of an information system or protected resources. 4. A process that checks the integrity of an entity." (IBM)</li> <li>2. Authentication: "1. In data security, the act of determining that a message has not been changed since leaving its point of origin. ... 4. In computer security, the act of identifying or verifying the eligibility of a station, originator, or individual to access specific categories of information." (Longley)</li> </ol>  |
| 3. | <p>budget</p> <p>193.1</p> | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "'Budgets' 308 shown in FIG. 5B are a special type of 'method' 1000 that may specify, among other things, limitations on usage of information content 304, and how usage will be paid for. Budgets 308 can specify, for example, how much of the total information content 304 can be used and/or copied. The methods 310 may prevent use of more than the amount specified by a specific budget." ('193 59:19-25) (See also Fig. 5B)</li> <li>2. "For example, consider the case of a security budget. One form of a typical budget might limit the user to 10Mb of decrypted data per month." ('193 265:9-11)</li> <li>3. "An example of the process steps used for the move of a budget record might look something like this: 1) Check the move budget (e.g., to determine the number of moves allowed)" ('193 265:24-27)</li> <li>4. "BUDGET method 408 may store budget information in a budget UDE..." ('193 182:25-26)</li> <li>5. "BUDGET method 408 may result in a 'budget remaining' field in a budget UDE being decremented by an amount specified by BILLING method 406." ('193 182:27-30)</li> <li>6. "In the preferred embodiment, a 'method' 1000 is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements and/or relationships for use in performing, and/or preparing a perform, basic instructions in relation to the operation of one or more electronic appliances 600." ('193 85:43-48; see also '193 136:20-25)</li> <li>7. "Budget process 408 limits how much content usage is permitted. For example, budget process 408 may limit the number of times content may be accessed or copied, or it may limit the number of pages or other amount of content that can be used based on, -for example, the number of dollars available in a credit account. Budget process 408 records and reports financial and other transaction information associated with such limits." ('193 58:27-34)</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>8. "BUDGET method 1510 may next perform a billing operation by adding a billing amount to a budget value (block 1602)." ('193 187:48-50)</p> <p>9. "The permissions and/or methods (i.e., budgets) carried by the portable appliance 2600 may have been assigned to it in conjunction with an 'encumbering' of another, stationary or other portable VDE electronic appliance 600." ('193 235:39-42)</p> <p>10. "Fields used for budget (but not for meter): 'Descending use counter ... Start date'" ('193 143:63 - 144:14)</p> <p>11. "A budget may be specified in dollars, deutsche marks, yen, and/or in any other monetary or content measurement schema and/or organization. The preferred embodiment output of the application, normally has three basic elements. A notation in the distribution portion of secure database 610 for each budget record created, the actual budget records, and a method option record for inclusion in a permissions record." ('193 265:44-51)</p> <p><b>Extrinsic:</b></p> <p>1. Budget: "A budget is the control mechanism for a meterable feature. A budget provides an upper limit for the volume of a meterable feature that a user (client) may use. Budgets consist of two values: a ceiling limit on use and an increment value that is added to the associated meter when a meterable event occurs. Budgets may be stand-alone or cascaded. A stand-alone budget only increments the meters for itself, while a cascaded budget can increment many meters from a single meterable event. A budget consists of an identification sextet, a descriptive area that describes the budget (cascade budget tuple and other miscellaneous flags), and a series of budget tuples. Each budget tuple consists of a budget and the increment value. It should be noted that a budget may be specified in meterable events or in dollars, based on the type of meter the budget will be compared against." (VDE ROI Device v1.0a, 2/9/94, IT00008582)</p> <p>2. Budget Object: "A governed element that defines the consumer's ability to provide payment using a specific payment type." (IT Glossary<sup>1</sup>, 1997-1998, ML00012B)</p> <p>3. Budget Object: "<i>An InterTrust system object</i> that defines the consumer's ability to provide payment using a specific payment type." (emphasis added) (IT System Developers Kit, 1997, TD00298C)</p> <p>4. Budget: "A control mechanism that limits operations on content based on billed amounts that can maintain a budget trail. A budget may be financially based (e.g., a number of dollars available for purchasing content use) or abstract (e.g. a total number of permitted usages)." (IT Glossary, 3/7/95, IT00709617)</p> <p>5. Budget: "*A fixed quantity of money, time, etc. against which the cost of operation is charged. Budget activities usually also involve reporting." (IT Glossary, 8/21/95, IT0032371)</p> |

<sup>1</sup> "IT Glossary" herein is a generic reference to several "glossaries" that have been created by InterTrust and that are further identified by Bates number and/or IT document number.

|    | Claim Term/Phrase                       | Evidence Supporting MS Construction  |
|----|---|--|
| 4. | clearinghouse<br><br>193.19             | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Distribution involves three types of entity. Creators usually are the source of distribution. They typically set the control structure 'context' and can control the rights which are passed into a distribution network. Distributors are users who form a link between object (content) end users and object (content) creators. They can provide a two-way conduit for rights and audit data. Clearinghouses may provide independent financial services, such as credit and/or billing services, and can serve as distributors and/or creators. Through a permissions and budgeting process, these parties collectively can establish fine control over the type and extent of rights usage and/or auditing activities." ('193 267:34-45)</li> <li>2. "Payment credit or currency may then be automatically communicated in protected (at least in part encrypted) form through telecommunication of a VDE container to an appropriate party such as a clearinghouse, provider of original property content or appliance, or an agent for such provider (other than a clearinghouse)." ('193 36:64 - 37:3)</li> <li>3. "...if appropriate credit (e.g. an electronic clearinghouse account from a clearinghouse such as VISA or AT&amp;T) is available..." ('193 25:22-24)</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>4. Clearinghouse: "*A facility that receives reports of content use and in turn reports payments and usage to content creators and distributors." (IT Glossary, 8/21/95, TD00068B, IT00032372)</li> </ol> |
| 5. | compares<br><br>900.155                 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "ROS 602 also provides a tagging and sequencing scheme that may be used within the loadable component assemblies 690 to detect tampering by substitution. Each element comprising a component assembly 690 may be loaded into an SPU 500, decrypted using encrypt/decrypt engine 522, and then tested/compared to ensure that the proper element has been loaded. Several independent comparisons may be used to ensure there has been no unauthorized substitution. For example, the public and private copies of the element ID may be compared to ensure that they are the same, thereby preventing gross substitution of elements." ('193 87:41-51)</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Compare: "1. To examine two items to discover their relative magnitudes, their relative positions in an order or in a sequence, or whether they are identical in given characteristics. 2. To examine two or more items for identity, similarity, equality, relative magnitude, or order in a sequence." (IBM)</li> <li>2. Comparison: "The process of examining two or more items for identity, similarity, equality, relative magnitude, or for order in sequence." (IBM)</li> </ol>  |
| 6. | component assembly<br><br>912.8, 912.35 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Many such load modules are inherently configurable, aggregatable, portable, and extensible and singularly, or in combination (along with associated data), run as control methods under the VDE transaction operating environment." ('193 25:48-52)</li> </ol>   |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>2. "Much of the functionality provided by ROS 602 in the preferred embodiment may be based on 'components' that can be securely, independently deliverable, replaceable and capable of being modified (e.g., under appropriately secure conditions and authorizations). Moreover, the 'components' may themselves be made of independently deliverable elements. ROS 602 may assemble these elements together (using a construct provided by the preferred embodiment called a 'channel') at execution time. For example, a 'load module' for execution by SPU 500 may reference one or more 'method cores,' method parameters and other associated data structures that ROS 602 may collect and assemble together to perform a task such as billing or metering. Different users may have different combinations of elements, and some of the elements may be customizable by users with appropriate authorization." ('193 77:12-27)</p> <p>3. "As discussed above, ROS 602 in the preferred embodiment is a component-based architecture. ROS VDE functions 604 may be based on segmented, independently loadable executable 'component assemblies' 690. These component assemblies 690 are independently securely deliverable. The component assemblies 690 provided by the preferred embodiment comprise code and data elements that are themselves independently deliverable. Thus, each component assembly 690 provided by the preferred embodiment is comprised of independently securely deliverable elements which may be communicated using VDE secure communication techniques, between VDE secure subsystems. These component assemblies 690 are the basic functional unit provided by ROS 602. The component assemblies 690 are executed to perform operating system or application tasks. Thus, some component assemblies 690 may be considered to be part of the ROS operating system 602, while other component assemblies may be considered to be 'applications' that run under the support of the operating system." ('193 83:11-22)</p> <p>4. "A complete VDE process to service a 'use event' may typically be constructed as a combination of methods 1000." ('193 181:20-21)</p> <p>5. "The audit information may be, in part, or whole, in some summary and/or analyzed form further processed at the clearinghouse and/or may be combined with other information to form a, at least in part, derived set of information and inserted into one or more at least in part secure VDE objects to be communicated to said one or more (further) auditor parties." ('193 272:29-36)</p> <p>6. "Components 690 are preferably designed to be easily separable and individually loadable. ROS 602 assembles these elements together into an executable component assembly 690 prior to loading and executing the component assembly (e.g., in a secure operating environment such as SPE 503 and/or HPE 655)." ('193 83:43-48)</p> <p>7. "component assemblies 690" ('193 83:23); see also "components 690" ('193 86:51-52)</p> <p>8. "In the preferred embodiment, ROS 602 assembles component assemblies 690 based on the following types of elements: Permissions Records ('PERC's) 808; Method 'Cores' 1000; Load Modules 1100; Data Elements (e.g., User Data Elements ('UDEs') 1200 and Method Data Elements ('MDEs') 1202); and Other component assemblies 690." ('193 85:21-29)</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>9. "...creation of component assemblies 690 from independently deliverable modules such as method cores 1000, load modules 1100, and data structures such as UDEs 1200." ('193 170:2-4)</p> <p>10. "ROS 602 also provides a tagging and sequencing scheme that may be used within the loadable component assemblies 690 to detect tampering by substitution. Each element comprising a component assembly 690 may be loaded into an SPU 500, decrypted using encrypt/decrypt engine 522, and then tested/compared to ensure that the proper element has been loaded. Several independent comparisons may be used to ensure there has been no unauthorized substitution. For example, the public and private copies of the element ID may be compared to ensure that they are the same, thereby preventing gross substitution of elements. In addition, a validation/correlation tag stored under the encrypted layer of the loadable element may be compared to make sure it matches one or more tags provided by a requesting process. This prevents unauthorized use of information. As a third protection, a device assigned tag (e.g., a sequence number) stored under an encryption layer of a loadable element may be checked to make sure it matches a corresponding tag value expected by SPU 500. This prevents substitution of older elements. Validation/correlation tags are typically passed only in secure wrappers to prevent plaintext exposure of this information outside of SPU 500." ('193 87:41-62)</p> <p>11. "Memory manager 578 and virtual memory manager 580 in the preferred embodiment manage ROM 532 and RAM 534 memory within SPU 500 in the preferred embodiment. Virtual memory manager 580 provides a fully 'virtual' memory system to increase the amount of 'virtual' RAM available in the SPE secure execution space beyond the amount of physical RAM 534a provided by SPU 500. Memory manager 578 manages the memory in the secure execution space, controlling how it is accessed, allocated and deallocated. SPU MMU 540, if present, supports virtual memory manager 580 and memory manager 578 in the preferred embodiment. In some 'minimal' configurations of SPU 500 there may be no virtual memory capability and all memory management functions will be handled by memory manager 578. Memory management can also be used to help enforce the security provided by SPE 503. In some classes of SPUs 500, for example, the kernel memory manager 578 may use hardware memory management unit (MMU) 540 to provide page level protection within the SPU 500. Such a hardware-based memory management system provides an effective mechanism for protecting VDE component assemblies 690 from compromise by 'rogue' load modules." ('193 109:24-45)</p> <p>12. "The channel 594 and its header 596 comprise a data structure that 'binds' or references elements of one or more component assemblies 690. Thus, the channel 594 is the mechanism in the preferred embodiment that collects together or assembles the elements shown in FIG. 11E into a component assembly 690 that may be used for event processing." ('193 115:65 - 116:4)</p> <p>13. "It reads the appropriate open control elements from the secure database (or the container, such as, for example, in the case of a traveling object), and 'binds' or 'links' these particular appropriate control elements together in order to control opening of the object for this user." ('193 185:42-46)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>14. "Thus, PERC 808 in effect contains a 'list of assembly instructions' or a 'plan' specifying what elements ROS 602 is to assemble together into a component assembly and how the elements are to be connected together. PERC 808 may itself contain data or other elements that are to become part of the component assembly 690." ('193 85:30-39)</p> <p>15. "The selected method event record 1012, in turn, specifies the appropriate information (e.g., load module(s) 1100, data element UDE(s) and MDE(s) 1200, 1202, and/or PERC(s) 808) used to construct a component assembly 690 for execution in response to the event that has occurred. ..." ('193 138:31-36)</p> <p>16. "As mentioned above, ROS 602 provides several layers of security to ensure the security of component assemblies 690. One important security layer involves ensuring that certain component assemblies 690 are formed, loaded and executed only in secure execution space such as provided within an SPU 500. Components 690 and/or elements comprising them may be stored on external media encrypted using local SPU 500 generated and/or distributor provided keys." ('193 87:33-40)</p> <p>17. "ROS 602 also provides a tagging and sequencing scheme that may be used within the loadable component assemblies 690 to detect tampering by substitution." ('193 87:41-43)</p> <p>18. "ROS 602 generates component assemblies 690 in a secure manner. As shown graphically in FIGS. 11I and 11J, the different elements comprising a component assembly 690 may be 'interlocking' in the sense that they can only go together in ways that are intended by the VDE participants who created the elements and/or specified the component assemblies. ROS 602 includes security protections that can prevent an unauthorized person from modifying elements, and also prevent an unauthorized person from substituting elements." ('193 84:60 - 85:2)</p> <p>19. "ROS 602 assembles these elements together into an executable component assembly 690 prior to loading and executing the component assembly (e.g., in a secure operating environment such as SPE 503 and/or HPE 655). ROS 602 provides an element identification and referencing mechanism that includes information necessary to automatically assemble elements into a component assembly 690 in a secure manner prior to, and/or during, execution." ('193 83:44-52)</p> <p>20. "Wherein said processor includes: retrieving means for retrieving at least one component, and at least one record that specifies a component assembly, from said memory devices, checking means coupled to said retrieving means for checking said component and/or said record for validity, and using means coupled to said retrieving means for using said component to form said component assembly in accordance with said record." ('107 Application p. 782 claim 80)</p> <p>21. "These called-for method(s) and data structure(s) (e.g., load modules 1100, UDEs 1200 and/or MDEs 1202) are each decrypted using encrypt/decrypt manager 556 (if necessary), and are then each validated using key and tag manager 558. Channel manager 562 constructs any necessary 'jump table' references to, in effect, 'link' or 'bind' the elements into a single cohesive executable so the load module(s) can reference data structures and any other load module(s) in the</p> |

|    | Claim<br>Term/Phrase                             | Evidence Supporting MS Construction  |
|----|--|--|
|    |  | <p>component assembly. Channel manager 562 may then issue calls to LMEM 568 to load the executable as an active task.” (‘193 116:25-35)</p> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Component: “1. Hardware or software that is part of a functional unit. 2. A functional part of an operating system. 3. A set of modules that performs a major function within a system.” (IBM)</li> <li>2. Component: “In data communications, a device or set of devices, consisting of hardware, along with its firmware, and or software that performs a specific function on a computer communications network. A Component is a part of a larger system, and may itself consist of other components.” (Longley)</li> <li>3. Record: “1. In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are call structures. 2. A set of data treated as a unit. 3. A set of one or more related data items grouped for processing.” (IBM)</li> <li>4. Record: “1. In computing, a collection of related data treated as a unit, e.g. details of name, address, age, occupation and department of an employee in a personnel file. 2. In computing, to store signals on a recording medium for later use.” (Longley)</li> <li>5. Record: “1. A collection of related data or words treated as a unit and saved in a position dependent fashion within a file or other such unit. 2. A set of data items, called fields, treated as a unit.” (Booth)</li> <li>6. Secure: “Pertaining to the control of who can use an object and to the extent to which the object can be used by controlling the authority given to the user.” (IBM)</li> </ol> |
| 7. | <p>contain</p> <p>683.2</p> <p>912.8, 912.35</p> | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “Container 300y may contain and/or reference rules and control information 300y(1) that specify the manner in which searching and routing information use and any changes may be paid for.” (‘193 241:36-39)</li> <li>2. “Each logical object structure 800 may also include a ‘private body’ 806 containing or referencing a set of methods 1000 (i.e., programs or procedures) that control use and distribution of the object 300.” (‘193 128:25-28)</li> <li>3. “Therefore, stationary object structure 850 does not contain a permissions record (PERC) 808; rather, this permissions record is supplied and/or delivered separately (e.g., at a different time, over a different path, and/or by a different party) to the appliance/installation 600.” (‘193 130:18-22)</li> <li>4. “The content portion of a logical object may be organized as information contained in, not contained in, or partially contained in one or more objects.” (‘193 127:8-19)</li> <li>5. “Container 302 may ‘contain’ items without those items actually being stored within the container. For example, the container 302 may reference items that are available elsewhere such as in other containers at remote sites. Container 302 may reference items available at different times or only during limited times. Some items may be too large to store within container 302. Items may, for example, be delivered to the user in the form of a ‘live feed’ of video at a certain</li> </ol>  |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>time. Even then, the container 302 'contains' the live feed (by reference) in this example." ('193 58:49-58)</p> <p>6. "Load modules 1100 may contain or reference other load modules." ('193 86:47-48)</p> <p>7. "PERC 808(k) defines, among other things, the 'assembly instructions' for component assembly 690(k), and may contain or reference parts of some or all of the components that are to be assembled to create a component assembly." ('193 87:3-6)</p> <p>8. "Alternatively, traveling object PERCs 808 may contain or reference budget records..." ('193 130:63-64)</p> <p>9. "Method 'core' 1000' in the preferred embodiment may contain or reference one or more data elements such as MDEs 1202 and UDEs 1200." ('193 136:32-34)</p> <p>10. "Container 300y may contain and/or reference rules and control information 300y(1) that specify the manner in which searching and routing information use and any changes may be paid for." ('193 241:36-39)</p> <p>11. "Trusted go-between 4700 registers the contract 4068, and then creates an electronic list of rules based on contract 4068. A partial example rule list is shown in FIG. 130A. Although the FIG. 130A conditions are shown as being written on a clipboard, in the preferred embodiment the" ('683 54:29-37)</p> <p>12. See also prior art referred to in the relevant InterTrust patent file histories, e.g. U.S. Patent No. 5,715,403</p> <p><b>Extrinsic:</b></p> <p>1. Container: "contains protected <i>content</i>, which is divided into one or more <i>atomic elements</i>, and, optionally, <i>PERCs</i> governing the <i>content</i> and may be manipulated only as specified by a <i>PERC</i>." (IT Glossary, 4/6/95, IT00028206)</p> <p>2. Container: "A packaging mechanism, consisting of: *One or more Element-derived components. *An organization mechanism which provides a unique name within a flat namespace for each of the components in a Container." (IT Glossary, 5/12/95, IT00028293)</p> <p>3. Container: "A protected digital information storage and transport mechanism for packaging content and control information." (IT Glossary, 8/21/95, TD00068B, IT00032372)</p> <p>4. Container: "A collection of content and control-related information." (IT VDE Container Overview, 2/10/95, ETM-9999 Version 0.21, IT00051228)</p> <p>5. Container: "A dynamic data structure, the elements of which are arbitrary data items whose type is not known when the program is written." (Que)</p> <p>6. Container: "Abstract data type storing a collection of objects (elements)." (Laplante)</p> <p>7. See also IT00037-44, IT002734-39, IT004188-96, IT0031572-85, IN00075960, IT00703055-71, IT0052146-64, IN00441189-224, IN0075983-87</p> <p>8. Contain: "In data security, a multilevel information structure. A container has a classification and may contain objects and/or other containers." (Longley)</p> <p>9. U.S. Patent No. 5,369,702</p> <p>10. See also Microsoft PLR 4-2 Exhs. E &amp; F as revised, and InterTrust's Rule 30(b)(6) testimony.</p> |

|    | Claim<br>Term/Phrase  | Evidence Supporting MS Construction  |
|----|---|--|
| 8. | control (n.)<br><br>193.1, 193.11,<br>193.15,<br>193.19<br>683.2<br>891.1 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Claims ... are allowable over the prior art of record. The instant claims provide for first and second entity or control or procedure or executable code that are separately, remotely and different from each to combine or process or execute an operation or procedure based on at least first and second control or procedure or executable code in an electronic appliance or secure operating environment or third party different and remote from the first and second entity or control or procedure or executable code." (08/964,333 Patent Application Prosecution History, Office Action, 9/22/98, p. 3 (MSI028945))</li> <li>2. "The virtual distribution environment 100 prevents use of protected information except as permitted by the 'rules and controls' (control information)." ('193 56:26-28)</li> <li>3. "As mentioned above, virtual distribution environment 100 'associates' content with corresponding 'rules and controls,' and prevents the content from being used or accessed unless a set of corresponding 'rules and controls' is available." ('193 57:18-22)</li> <li>4. "...at least one rule and/or control associated with the software agent that governs the agent's operation." ('193 241:2-3)</li> <li>5. "In this example control information may include one or more component assemblies that describe the articles within such a container (e.g. one or more event methods referencing map tables and/or algorithms that describe the extent of each article)." ('193 309:5-9)</li> <li>6. "Even if a consumer has a copy of a video program, she cannot watch or copy the program unless she has 'rules and controls' that authorize use of the program. She can use the program only as permitted by the 'rules and controls.'" ('193 53:60-63)</li> <li>7. "A control set 914 contains a list of required methods that must be used to exercise a specific right (i.e., process events associated with a right)." ('193 151:14-16)</li> <li>8. "If necessary, trusted go-between 4700 may obtain and register any methods, rules and/or controls it needs to use or manipulate the object 300 and/or its contents (FIG. 122 block 4778)." ('683, 47:42-45)</li> <li>9. "These rights govern use of the VDE object 300 by that user or user group. For instance, the user may have an 'access' right, and an 'extraction' right, but not a 'copy' right." ('193 159:23-26)</li> <li>10. "To provide for this, ROS 602 may include a 'redirector' 684 that allows such 'non-VDE aware' applications 608(b) to access VDE objects 300 and functions 604. Redirector 684, in the preferred embodiment, translates OS calls directed to the 'other OS functions' 606 into calls to the 'VDE functions' 604. As one simple example, redirector 684 may intercept a 'file open' call from application 608(b), determine whether the file to be opened is contained within a VDE container 300, and if it is, generate appropriate VDE function call(s) to file system 687 to open the VDE container (and potentially generate events to HPE 655 and/or SPE 503 to determine the name(s) of file(s) that may be stored in a VDE object 300, establish a control structure associated with a VDE object 300, perform a registration for a VDE object 300, etc.). Without redirector 684 in this example, a</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>non-VDE aware application such as 608b could access only the part of API 682 that provides an interface to other OS functions 606, and therefore could not access any VDE functions.” (‘193 82:27-45)</p> <p>11. “An executing process cannot access memory outside its domain and can only communicate with other processes through services provided by and mediated by privileged kernel/dispatcher software 552 within the SPU 500.” (‘193 109:53-57)</p> <p>12. “An electronic appliance 600 may not access an object unless a corresponding PERC 808 is present, and may only use the object and related information as permitted by the control structures contained within the PERC.” (‘193 118:17-31)</p> <p>13. “Load modules are not necessarily directly governed by PERCs 808 that control them, nor must they contain any time/date information or expiration dates. The only control consideration in the preferred embodiment is that one or more methods 1000 reference them using a correlation tag (the value of a protected object created by the load module’s owner, distributed to authorized parties for inclusion in their methods, and to which access and use is controlled by one or more PERCs 808). If a method core 1000’ references a load module 1100 and asserts the proper correlation tag (and the load module satisfies the internal tamper checks for the SPE 503), then that load module can be loaded and executed, or it can be acquired from, shipped to, updated, or deleted by, other systems.” (193 139:60 - 140:6)</p> <p>14. “In the preferred embodiment, SPE RPC manager 550 first references a service request against the RPC service table to determine the location of the service manager that may service the request. The RPC manager 550 then routes the service request to the appropriate service manager for action. Service requests are handled by the service manager within the SPE 503 using the RPC dispatch table to dispatch the request. Once the RPC manager 550 locates the service reference in the RPC dispatch table, the load module that services the request is called and loaded using the load module execution manager 568. The load module execution manager 568 passes control to the requested load module after performing all required context configuration, or if necessary may first issue a request to load it from the external management files 610.” (‘193 148:55-58)</p> <p>15. “Although methods 1000 can have virtually unlimited variety and some may even be user-defined, certain basic ‘use’ type methods are preferably used in the preferred embodiment to control most of the more fundamental object manipulation and other functions provided by VDE 100. For example, the following high level methods would typically be provided for object manipulation: OPEN method, READ method, WRITE method, CLOSE method. An OPEN method is used to control opening a container so its contents may be accessed. A READ method is used to control the access to contents in a container. A WRITE method is used to control the insertion of contents into a container. A CLOSE method is used to close a container that has been opened.” (‘193 183:12-29)</p> <p>16. “FIG. 54 is a flowchart of an example of program control steps performed by an ACCESS method 2000. As described above, an ACCESS method may be used to access content embedded in an object 300 so it can be written to, read from, or</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>otherwise manipulated or processed. In many cases, the ACCESS method may be relatively trivial since the object may, for example, be stored in a local storage that is easily accessible. However, in the general case, an ACCESS method 2000 must go through a more complicated procedure in order to obtain the object. For example, some objects (or parts of objects) may only be available at remote sites or may be provided in the form of a real-time download or feed (e.g., in the case of broadcast transmissions). Even if the object is stored locally to the VDE node, it may be stored as a secure or protected object so that it is not directly accessible to a calling process. ACCESS method 2000 establishes the connections, routings, and security requisites needed to access the object. These steps may be performed transparently to the calling process so that the calling process only needs to issue an access request and the particular ACCESS method corresponding to the object or class of objects handles all of the details and logistics involved in actually accessing the object.” (‘193 188:59-67)</p> <p>17. “The READ control method 1652 must determine which key to use to decrypt content if it is going to release decrypted content to the user (block 1758). READ control method 1652 may make this key determination based, in part, upon the PERC 808 for the object (block 1760). READ control method 1652 may then call an ACCESS method to actually obtain the encrypted content to be decrypted (block 1762). The content is then decrypted using the key determined by block 1758 (block 1764).” (‘193 192:2-24)</p> <p>18. See also prior art referred to in the relevant InterTrust patent file histories, e.g., references made at the following bates ranges: MSI026598-602, MSI26626-7, MSI26630-42; MSI028808-11, MSI28846-52, MSI28728-62, MSI28857-58, MSI28944-97, MSI28953-56</p> <p>19. “C<sub>c</sub> may further include, for example: (a) a requirement that distributors ensure that creator C receive \$1 per article accessed by users and/or user/distributors, which payment allows a user to access such an article for a period of no more than six months (e.g. using a map-type meter method that is aged once per month, time aged decryption keys, expiration dates associated with relevant permissions records, etc.” (‘193 309:10-16)</p> <p>20. “It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g, summary, table of contents) and secured content control information which ensures the performance of control information.” (‘193 15:41-46)</p> <p>21. “Because of the breadth of issues resolved by the present invention, it can provide the emerging ‘electronic highway’ with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions.” (‘193 17:22-28)</p> <p>22. “... (as allowed, or not prevented, by senior control information).” (‘193 303:67 - 304:1)</p> <p>23. “For purposes of expedition, applicants are rewriting these dependent claims into independent form, .... In addition, applicants have ... replaced ‘necessary in order to gain’ with ‘allowing’ in now-cancelled claim 204 incorporated into formerly dependent claims 209 &amp; 21 1 [issued claim 35] ....” (Prosecution</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>History for the 08/780,545 Patent Application (issued as the '912), Amendment, 10/29/98)</p> <p>24. "VDE can: (a) audit and analyze the use of content, (b) ensure that content is used only in authorized ways, and (c) allow information regarding content usage to be used only in ways approved by content users." ('193 4:51-56)</p> <p>25. "VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties." ('193 6:33-34)</p> <p>26. "VDE ensures that certain prerequisites necessary for a given transaction to occur are met." ('193 20:27-28)</p> <p><b>Extrinsic:</b></p> <p>1. Control: "The determination of the time and order in which the parts of a data processing system and the devices that contain those parts perform the input, processing, storage, and output functions." (IBM)</p> <p>2. "5. Control Notes ... A Control must execute as a transaction ... A Control may require pre-conditions – that is that one or more other Controls have been executed before the Control is executed. ... 7. Control Execution Flow The following pseudocode describes the approximate execution sequence for a View Control ... 8. Operation of a Control (Execution of 'Rules and Consequences') ..." (VDE Controls Notes, IT00051953-55)</p> <p>3. Control: "A business rule that governs the use of content." (IT Glossary, 1997-1998, ML00012B)</p> <p>4. Control: "A set of rules and consequences that apply to a governed element. The term control can apply to either a control program or a control set." (IT Glossary, 1997-2000, ML00012D)</p> <p>5. Control: "<i>*Control Element</i>: A data structure that governs [sic] the operation of a control mechanism (e.g., meter element, budget element, report element, trail element). <i>*Control mechanism</i>: One of the mechanisms that controls and performs operations on a VDE object (e.g. meter, bill, budget). A control mechanism is distinct from a control element in that it specifies the execution of some process. <i>*Control object</i>: A data structure that is used to implement some VDE control: a PERC, a control element, a control parameter, or the data representing a control mechanism. <i>*Control Parameter</i>: A data structure that is input to a control mechanism and that serves as part of the mechanism's specifications. For example, a billing mechanism might have a pricing parameter; a creator using that mechanism could alter the parameter but not change the mechanism itself." (IT Glossary, 3/7/95, IT00709618)</p> <p>6. Control: "Defines rules and consequences for operations on a Property Chunk. A Control may be implemented by a process of arbitrary complexity (within the limits posed by the capability of the Node." (IT Glossary, 5/12/95, IT00028293)</p> <p>7. Control: "A set of rules and consequences for operations on content, such as pricing, payment models, usage reporting etc." (IT Glossary, 8/21/95, TD00068B, IT00032373)</p> <p>8. Control: "An object of the InterTrust Commerce Architecture that specifies business rules. Controls are applied at any time and at any point in the Chain of</p> |



|    | Claim Term/Phrase                                | Evidence Supporting MS Construction  |
|----|--|--|
|    |  | <p>Handling and Control. InterTrust controls are dynamic, independent, and persistent.” (IT Glossary, 11/17/96, TD00189J, IT00035865)</p> <p>9. “‘Rules and Controls’ means any electronic information that directs, enables, specifies, describes, and/or provides contributing means for performing or not-performing, permitted and/or required operations related to Content, including, for example, restricting or otherwise governing the performance of operations, such as, for example, Management of such Content.” (License Agreement, InterTrust/Universal Music Group, 4/13/99, Exhibit 11 to InterTrust 30(b)(6))</p> <p>10. “A set of control elements corresponding to all of the property elements of a property. There may be zero or more controls for a given property.” (IT 0028204)</p> <p>11. “CONTROL(S): Controls refer to the rules and consequences associated with DigiBox containers. Controls may be applied dynamically...” (IT00035961)</p> <p>12. “CONTROL: The rules associated with a governed entity such as a DigiBox container, property, or another control ... applied dynamically. InterTrust controls are dynamic, independent, and persistent.” (IT00035920)</p> <p>13. “... controls implement business rules...” (IT00035892)</p> <p>14. “The function of performing required operations when certain specific conditions occur or when interpreting and acting upon instructions.” (Webster’s)</p> <p>15. Access (n.): “2. The use of an access method. 3. The manner in which files or data sets are referred to by the computer. ... 5. In computer security, a specific type of interaction between a subject and an object that results in the flow of information from one to the other.” (IBM)</p> <p>16. Access (n.): “1. In access control, a specific type of interaction between a subject and an object that results in the flow of information from one to the other ... 3. In computing, the manner in which files or data sets are referred to by a computer.” (Longley)</p> <p>17. Access(ing) (v.): “1. To obtain the use of a computer resource. ... 4. To obtain data from or to put data in storage.” (IBM)</p> <p>18. Least privilege: “Each user and each program should operate using the fewest privileges possible. In this way, the damage from an inadvertent or malicious attack is minimized.” (Pfleeger)</p> <p>19. See also IT00125, IT31410-14, IT703083-89, IT51721-26, IT00735936 (key), IT51956 et seq., IN0075983-87, IN0075989-93</p> <p>20. See also Microsoft PLR 4-2 Exhs. E &amp; F as revised, and InterTrust’s Rule 30(b)(6) testimony.</p> |
| 9. | controlling, control (v.)<br><br>193.1<br>861.58 | <p><b>Intrinsic:</b></p> <p>1. “ROS 602 includes software intended for execution by SPU microprocessor 520 for, in part, controlling usage of VDE related objects 300 by electronic appliance 600. As will be explained, these SPU programs include ‘load modules’ for performing basic control functions.” (‘193 66:5-8)</p> <p>2. “VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information.” (‘193 11:60-63)</p> <p>3. “It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information</p>   |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>(e.g., summary, table of contents) and secured content control information which ensures the performance of control information." ('193 15:41-46)</p> <ol style="list-style-type: none"> <li>4. "VDE ensures that certain prerequisites necessary for a given transaction to occur are met." ('193 20:27-28)</li> <li>5. "The virtual distribution environment 100 prevents use of protected information except as permitted by the 'rules and controls' (control information)." ('193 56:26-28)</li> <li>6. "As mentioned above, virtual distribution environment 100 'associates' content with corresponding 'rules and controls,' and prevents the content from being used or accessed unless a set of corresponding 'rules and controls' is available." ('193 57:18-22)</li> <li>7. "VDE can: (a) audit and analyze the use of content, (b) ensure that content is used only in authorized ways, and (c) allow information regarding content usage to be used only in ways approved by content users." ('193 4:51-56)</li> <li>8. "VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties." ('193 6:33-35)</li> <li>9. "It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g., summary, table of contents) and secured content control information which ensures the performance of control information." ('193 15:41-46)</li> <li>10. "Because of the breadth of issues resolved by the present invention, it can provide the emerging 'electronic highway' with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions." ('193 17:22-28)</li> <li>11. "VDE ensures that certain prerequisites necessary for a given transaction to occur are met." ('193 20:27-28)</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Control: "The determination of the time and order in which the parts of a data processing system and the devices that contain those parts perform the input, processing, storage, and output functions." (IBM)</li> <li>2. Control: "A business rule that governs the use of content." (IT Glossary, 1997-1998, ML00012B)</li> <li>3. Control: "A set of rules and consequences that apply to a governed element. The term control can apply to either a control program or a control set." (IT Glossary, 1997-2000, ML00012D)</li> <li>4. Control: "<i>*Control Element:</i> A data structure that giverns (<i>sic</i>) the operation of a control mechanism (e.g., meter element, budget element, report element, trail element). <i>*Control mechanism:</i> One of the mechanisms that controls and performs operations on a VDE object (e.g. meter, bill, budget). A control mechanism is distinct from a control element in that it specifies the execution of some process. <i>*Control object:</i> A data structure that is used to implement some VDE control: a PERC, a control element, a control parameter, or the data representing a control mechanism. <i>*Control Parameter:</i> A data structure that is</li> </ol> |

|     | Claim<br>Term/Phrase  | Evidence Supporting MS Construction  |
|-----|---|--|
|     |   | <p>input to a control mechanism and that serves as part of the mechanism's specifications. For example, a billing mechanism might have a pricing parameter; a creator using that mechanism could alter the parameter but not change the mechanism itself." (IT Glossary, 3/7/95, IT00709618)</p> <p>5. Control: "Defines rules and consequences for operations on a Property Chunk. A Control may be implemented by a process of arbitrary complexity (within the limits posed by the capability of the Node." (IT Glossary, 5/12/95, IT00028293)</p> <p>6. Control: "A set of rules and consequences for operations on content, such as pricing, payment models, usage reporting etc." (IT Glossary, 8/21/95, TD00068B, IT00032373)</p>   |
| 10. | <p>copy, copied, copying</p> <p>193.1, 193.11, 193.15, 193.19</p> | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "These rights govern use of the VDE object 300 by that user or user group. For instance, the user may have an 'access' right, and an 'extraction' right, but not a 'copy' right." ('193 159:23-26)</li> <li>2. "At the same time, electronic testing will allow users to receive a copy (encrypted or unencrypted) of their test results when they leave the test sessions." ('193 319:12-15)</li> <li>3. "This is because VDE objects may contain data that can be electronically copied outside the confines of a VDE node. If the content is encrypted, the copies will also be encrypted and the copier cannot gain access to the content unless she has the appropriate decryption key(s)." ('193 129:3-8)</li> <li>27. "Even if a consumer has a copy of a video program, she cannot watch or copy the program unless she has 'rules and controls' that authorize use of the program. She can use the program only as permitted by the 'rules and controls.'" ('193 53:60-63)</li> <li>4. "For example, if a software program was distributed as a traveling object, a user of the program who wished to supply it or a usable copy of it to a friend would normally be free to do so." ('193 131:65 - 132:1)</li> <li>5. "Storing a first digital file and a first control in a first secure container, said first control constituting a first budget which governs the number of copies which may be made of said first digital file or a portion of said first digital file while said first digital file is contained in said first secure container." ('193 330:1 -331:25 (claim 60))</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Copy: "A product of a document copying process." (IBM)</li> </ol> |
| 11. | <p>derive</p> <p>900.155</p>                                      | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Such control information can continue to manage usage of container content if the container is 'embedded' into another VDE managed object, such as an object which contains plural embedded VDE containers, each of which contains content derived (extracted) from a different source." ('193 28:60-65)</li> </ol>  |
| 12. | <p>designating</p> <p>721.1</p>                                   |  |

|     | Claim Term/Phrase                                 | Evidence Supporting MS Construction  |
|-----|---|--|
| 13. | device class<br>721.1                             | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Furthermore, Applicants respectfully submit that some of the terms cited by the Examiner as 'indefinite' are either well-known by persons skilled in the art or inherently clear. For example, in Claims 1-4, 22-25, the term 'class' is used as part of the phrase 'device class.' Applicants respectfully submit that 'device class' is inherently clear, meaning a group of devices which share at least one attribute." (Prosecution History for the 08/689,754 Patent Application (issued as the '721), Amendment, 4/14/99, p. 14)</li> </ol> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Device: "1. A mechanical, electrical, or electronic contrivance with a specific purpose." (IBM)</li> <li>2. Device class: "The generic name for a group of device types." (IBM)</li> <li>3. Device type: "1. The name for a kind of device sharing the same model number; for example, 2311, 2400, 2400-1. Contrast with device class. 2. The generic name for a group of devices; for example, 5219 for IBM 5219 Printers. Contrast with device class." (IBM)</li> </ol>  |
| 14. | digital signature, digitally signing<br><br>721.1 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "There exist many well known processes for creating digital signatures. One example is the Digital Signature Algorithm (DSA). DSA uses a public-key signature scheme that performs a pair of transformations to generate and verify a digital value called a 'signature.'" ('721 10:60-64)</li> <li>2. "A verifying authority digitally 'signs' and 'certifies' those load modules or other executables it has verified (using a public key based digital signature and/or certificate based thereon, for example)." ('721 4:64-67)</li> <li>3. "The algorithm also makes use of a one-way hash function, H(m), such as, for example, the Secure Hash Algorithm. The first three parameters, p, q, and g, are public and may be shared across a network of users. The private key is x; the public key is y. To sign a message, m, using DSA, a signer generates a random number, k, less than q. The signer also generates: <math>r = (g^k \text{ mod } p) \text{ mod } q</math>; and <math>s = (k^{-1} (H(m) + xr)) \text{ mod } q</math>. The parameters r and s comprise the signer's signature, which may be sent to a recipient or distributed across a network." ('721 11:7-22)</li> <li>4. "Protected processing environment 108 then decrypts digital signature 106 using the second key 124--i.e., it opens strongbox 118 to retrieve the message digest 116 a verifying authority 100 placed in there. Protected processing environment 108 compares the version of message digest 116 it obtains from the digital signature 106 with the version of message digest 116' it calculates itself from load module 54 using the one way hash transformation 115. The message digests 116, 116' should be identical. If they do not match, digital signature 106 is not authentic or load module 54 has been changed--and protected processing environment 108 rejects load module 54." ('721 14:49-60)</li> <li>5. "One digital signature 106(1) can be created by encrypting message digest 116 with a 'private' key 122(1), another (different) digital signature 106(2) can be created by encrypting the message digest 116 with a different 'private' key 122(2), possibly employing a different signature algorithm." ('721 14:64 - 15:2)</li> <li>6. "Certificates play an important role in the trustedness of digital signatures, and</li> </ol> |

|   | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
|---|----------------------|--|-----------------------------|----------------------|-----------------------------|---|------|--|-------------------|--------------------|---|------------------------|----|---------------------------------|-------------------------|----|--|--------------------|--------|-----|----------------------|--------|-----|--|------|-----|----------------------------|------|-----|
|   |                      | <p>also are important in the public-key authentication communications protocol (to be discussed below). In the preferred embodiment, these certificates may include information about the trustedness/level of security of a particular VDE electronic appliance 600 (e.g., whether or not it has a hardware-based SPE 503 or is instead a less trusted software emulation type HPE 655) that can be used to avoid transmitting certain highly secure information to less trusted/secure VDE installations.” (‘193 203:58-67)</p> <p>7. “Master Keys: A ‘master’ key is a key used to encrypt other keys. An initial or ‘master’ key may be provided within PPE 650 for communicating other keys in a secure way. During initialization of PPE 650, code and shared keys are downloaded to the PPE. Since the code contains secure convolution algorithms and/or coefficients, it is comparable to a ‘master key.’ The shared keys may also be considered ‘master keys.’” (‘193 212:12-18)</p> <p>8. “FIGS. 64 through 67 illustrate the preferred public-key embodiment, but may also be used to help understand the secret-key versions. In secret-key embodiments, the certification process and the public key encryptions/decryptions are replaced with private-key encryptions, and the public key/private-key pairs are replaced with individual secret keys that are shared between the PPE 650 instance and the other parties (e.g., the load module supplier(s), the PPE manufacturer). In addition, the certificate generation process 2804 is not performed in secret-key embodiments, and no site identity certificates 2823 or VDE certificate database 2830 exist.” (‘193 211:18-30)</p> <p>9. “Key Types</p> <p>The detailed descriptions of key types below further explain secret-key embodiments; this summary is not intended as a complete description. The preferred embodiment PPE 650 can use different types of keys and/or different ‘shared secrets’ for different purposes. Some key types apply to a Public-Key/Secret Key implementation, other keys apply to a Secret Key only implementation, and still other key types apply to both. The following table lists examples of various key and ‘shared secret’ information used in the preferred embodiment, and where this information is used and stored:</p> <table> <tr> <th data-bbox="407 1318 812 1346">Key/Secret Information Type</th><th data-bbox="812 1318 998 1346">Used in PK or Non-PK</th><th data-bbox="998 1318 1404 1346">Example Storage Location(s)</th></tr> <tr> <td data-bbox="407 1346 812 1457">Master Key(s) (may include some of the specific keys mentioned below)</td><td data-bbox="812 1346 998 1457">Both</td><td data-bbox="998 1346 1404 1457">PPE<br/>Manufacturing facility<br/>VDE administrator</td></tr> <tr> <td data-bbox="407 1457 812 1520">Manufacturing Key</td><td data-bbox="812 1457 998 1520">Both (PK optional)</td><td data-bbox="998 1457 1404 1520">PPE (PK case)<br/>Manufacturing facility</td></tr> <tr> <td data-bbox="407 1520 812 1583">Certification key pair</td><td data-bbox="812 1520 998 1583">PK</td><td data-bbox="998 1520 1404 1583">PPE<br/>Certification repository</td></tr> <tr> <td data-bbox="407 1583 812 1667">Public/private key pair</td><td data-bbox="812 1583 998 1667">PK</td><td data-bbox="998 1583 1404 1667">PPE<br/>Certification repository<br/>(Public Key only)</td></tr> <tr> <td data-bbox="407 1667 812 1694">Initial secret key</td><td data-bbox="812 1667 998 1694">Non-PK</td><td data-bbox="998 1667 1404 1694">PPE</td></tr> <tr> <td data-bbox="407 1694 812 1722">PPE manufacturing ID</td><td data-bbox="812 1694 998 1722">Non-PK</td><td data-bbox="998 1694 1404 1722">PPE</td></tr> <tr> <td data-bbox="407 1722 812 1782">Site ID, shared code, shared keys and shared secrets</td><td data-bbox="812 1722 998 1782">Both</td><td data-bbox="998 1722 1404 1782">PPE</td></tr> <tr> <td data-bbox="407 1782 812 1810">Download authorization key</td><td data-bbox="812 1782 998 1810">Both</td><td data-bbox="998 1782 1404 1810">PPE</td></tr> </table> | Key/Secret Information Type | Used in PK or Non-PK | Example Storage Location(s) | Master Key(s) (may include some of the specific keys mentioned below) | Both | PPE<br>Manufacturing facility<br>VDE administrator | Manufacturing Key | Both (PK optional) | PPE (PK case)<br>Manufacturing facility | Certification key pair | PK | PPE<br>Certification repository | Public/private key pair | PK | PPE<br>Certification repository<br>(Public Key only) | Initial secret key | Non-PK | PPE | PPE manufacturing ID | Non-PK | PPE | Site ID, shared code, shared keys and shared secrets | Both | PPE | Download authorization key | Both | PPE |
| Key/Secret Information Type   | Used in PK or Non-PK | Example Storage Location(s)  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Master Key(s) (may include some of the specific keys mentioned below) | Both                 | PPE<br>Manufacturing facility<br>VDE administrator   |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Manufacturing Key   | Both (PK optional)   | PPE (PK case)<br>Manufacturing facility  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Certification key pair  | PK                   | PPE<br>Certification repository  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Public/private key pair   | PK                   | PPE<br>Certification repository<br>(Public Key only)   |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Initial secret key  | Non-PK               | PPE  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| PPE manufacturing ID  | Non-PK               | PPE  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Site ID, shared code, shared keys and shared secrets                  | Both                 | PPE  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |
| Download authorization key  | Both                 | PPE  |                             |                      |                             |   |      |  |                   |                    |   |                        |    |                                 |                         |    |  |                    |        |     |                      |        |     |  |      |     |                            |      |     |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |  |  |
|--|----------------------|--|--|--|
|  |                      | <p>External communication keys and other info      Both</p> <p>Administrative object keys      Both</p> <p>Stationary object keys      Both</p> <p>Traveling object shared keys      Both</p> <p>Secure database keys      Both</p> <p>Private body keys      Both</p> <p>Content keys      Both</p> <p>Authorization shared secrets      Both</p> <p>Secure Database Back up keys      Both</p>   |  | <p>VDE administrator</p> <p>PPE</p> <p>Secure Database</p> <p>Permission record</p> <p>Permission record</p> <p>Permission record</p> <p>PPE</p> <p>Secure database</p> <p>Some objects</p> <p>Secure database</p> <p>Some objects</p> <p>Permission record</p> <p>PPE</p> <p>Secure database"</p> |
|  |                      | <p>(‘193 211:31 – 212:11)</p> <p>10. "The process for this selection is similar to the process used by EVENT methods to map events into atomic element numbers. DECRYPT method 2030 may then access an appropriate PERC 808 from the secure database 610 and loads a key (or 'seed') from a PERC (blocks 2034, 2036). This key information may be the actual decryption key to be used to decrypt the content, or it may be information from which the decryption key may be at least in part derived or calculated. If necessary, DECRYPT method 2030 computes the decryption key based on the information read from PERC 808 at block 2034 (block 2038). DECRYPT method 2030 then uses the obtained and/or calculated decryption key to actually decrypt the block of encrypted information (block 2040). DECRYPT method 2030 outputs the decrypted block (or the pointer indicating where it may be found), and terminates (termination point 2042)." (‘193 193:8-23)</p> <p>11. "A 'time aged key' in the preferred embodiment is not a 'true key' that can be used for encryption/decryption, but rather is a piece of information that a PPE 650, in conjunction with other information, can use to generate a 'true key.' This other information can be time-based, based on the particular 'ID' of the PPE 650, or both. Because the 'true key' is never exposed but is always generated within a secure PPE 650 environment, and because secure PPEs are required to generate the 'true key,' VDE 100 can use 'time aged' keys to significantly enhance security and flexibility of the system." (‘193 207:50-60)</p> <p>12. "Running the function with a time-aged key and inappropriate time values typically yields a useless key that will not decrypt." (‘193 208:38-40)</p> <p><b>Extrinsic:</b></p> <p>1. Digital Signature: "In computer security, encrypted data, appended to or part of a message, that enables a recipient to prove the identity of the sender." (IBM)</p> <p>2. Digital Signature: "1. In authentication, data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery. 2. In authentication, a data block appended to a message, or a complete encrypted message, such that the recipient can authenticate the message contents and/or prove that it could only have originated with the purported sender." (Longley)</p> |  |  |

|     | Claim Term/Phrase  | Evidence Supporting MS Construction  |
|-----|--|--|
|     |  | <p>3. "Let B be the recipient of a message M signed by A, then A's [digital] signature must satisfy three requirements: B must be able to validate A's signature on M. It must be impossible for anyone, including B, to forge A's signature. In case A should disavow signing a message M, it must be possible for a judge or third party to resolve a dispute arising between A and B. A digital signature therefore establishes sender authenticity ..., it also establishes data authenticity." (Denning, p. 14)</p> <p>4. "A cipher is unconditionally secure if, no matter how much ciphertext is intercepted, there is not enough information in the ciphertext to determine the plaintext uniquely." (Denning, p. 5) (Davies, pp. 41, 380)</p> <p>5. "A cipher is computationally secure, or strong, if it cannot be broken by systematic analysis with available resources." (Denning, p. 5) (Davies, pp. 41, 370)</p> <p>6. Key: "7. In computer security, a sequence of symbols used with a cryptographic algorithm for encrypting or decrypting data." (IBM)</p> <p>7. Key: "1. In cryptography, a sequence of symbols that controls the operations of encipherment and decipherment. 2. In cryptography, a symbol or sequence of symbols (or electrical or mechanical correlates of symbols) that control the operations of encryption and decryption)." (Longley)</p>  |
| 15. | <p>executable programming, executable</p> <p>721.34</p> <p>912.8, 912.35</p> | <p><b>Intrinsic:</b></p> <p>1. "Furthermore, applicants' independent claims 16, 36, 37 and 64 require secure delivery and use of plural executable items. See claim 16 ('securely delivering a first procedure ... securely delivering ... a second procedure separable or separate from said first procedure...'); claim 36 ('securely delivering plural executable procedures ...'), claim 37 ('securely delivering a first piece of executable code ... securely delivering a second piece of executable code ...') and claim 64 ('securely receiving a first load module ... securely receiving a second load module ...'). These features are not taught or suggested by either Rosen or Johnson. Johnson's databases comprise data, not executable code." (Prosecution for the 08/388,107, Patent Application, Amendment, 6/20/97, pp. 24-25) (MSI028848-49)</p> <p>2. "In addition, Applicants would like to draw the Examiner's attention to other sections of the specification in support of words or phrases cited by the Examiner as 'indefinite.' ... The noun 'executable,' as used in Claims ... 34-36 ..., is defined in the specification on page 7." (Prosecution History for the 08/689,754 Patent Application (issued as the '721 patent), Amendment, 4/14/99, pp. 13-14) (p. 7 of the original specification is '721 2:62 - 3:13 of the issued patent)</p> <p><b>Extrinsic:</b></p> <p>1. Execute: "1. To perform the actions specified by a program or a portion of a program." (IBM)</p> <p>2. Executable Program: "1. A program that has been link-edited and therefore can be run in a processor. 2. The set of machine language instructions that constitute the output from the compilation of a source program." (IBM)</p> |

|     | Claim<br>Term/Phrase                             | Evidence Supporting MS Construction   |
|-----|--|---|
| 16. | host<br>processing<br>environment<br><br>900.155 | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "Portions of ROS 602 in particular may desirably be included in ROM 658 (e.g., 'bootstrap' routines, POST routines, etc. for use in establishing an operating environment for electronic appliance 600 when power is applied)." ('193 63:13-17)</li> <li>2. "In the preferred embodiment, HPE 655 is a secure processing environment supported by a processor other than an SPU, such as for example an electronic appliance CPU 654 general-purpose microprocessor or other processing system or device. In the preferred embodiment, HPE 655 may be considered to 'emulate' an SPU 500 in the sense that it may use software to provide some or all of the processing resources provided in hardware and/or firmware by an SPU." ('193 79:60-67)</li> <li>3. "However, in applications where lesser security can be tolerated and/or the cost of an SPU 500 cannot be tolerated, the SPE 503 may be omitted and all secure processing may instead be performed by one or more secure HPEs 655 executing on general-purpose CPUs 654." ('193 81:4-8)</li> <li>4. "Integrity of Software-Based PPE Security: As discussed above in connection with FIG. 10, some applications may use a software-based protected processing environment 650 (such as a 'host event processing environment' (HPE) 655) providing a software-based tamper resistant barrier 674." ('900 230:57-61)</li> <li>5. "In one example, the software distribution medium 3370 might include installation materials 3470 and operational materials 3472. The installation materials 3470 may be executed by computer 3372 to install the operational materials 3472 onto the computer's hard disk 3376. The computer 3372 may then execute the operational materials 3472 from its hard disk 3376 to provide software-based protected processing environment 650 and associated software-based tamper resistant barrier 672." ('900 231:23-31)</li> <li>6. "The operational materials 3472 may provide executable code and associated data structures for providing protected processing environment 650 and associated software-based tamper resistant barrier 674." ('900 236:50-53)</li> <li>7. "HPE(s) 655 and SPE(s) 503 are self-contained computing and processing environments that may include their own operating system kernel 688 including code and data processing resources." ('193 79:36-39)</li> <li>8. "HPEs 655 may be provided in two types: secure and not secure." ('193 80:8-9)</li> <li>9. "[T]his example also includes one or more Host Event Processing Environments ('HPEs') 655 and/or one or more Secure Event Processing Environment ('SPEs') 503 (these environments may be generically referred to as 'Protected Processing Environments' 650)." ('193 79:31-35)</li> <li>10. "HPEs 655 may (as shown in FIG. 10) be provided with a software-based tamper resistant barrier 674 that makes them more secure. Such a software-based tamper resistant barrier 674 may be created by software executing on general-purpose CPU 654. Such a 'secure' HPE 655 can be used by ROS 602 to execute processes that, while still needing security, may not require the degree of security provided by SPU 500. This can be especially beneficial in architectures providing both an SPE 503 and an HPE 655. The SPU 502 may be used to perform all truly secure processing, whereas one or more HPEs 655 may be used to provide additional</li> </ol> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>secure (albeit possibly less secure than the SPE) processing using host processor or other general purpose resources that may be available within an electronic appliance 600. Any service may be provided by such a secure HPE 655" ('193 80:22-36)</p> <p>11. "The software-based tamper resistant barrier 674 provided by HPE 655 may be provided, for example, by: introducing time checks and/or code modifications to complicate the process of stepping through code comprising a portion of kernel 688a and/or a portion of component assemblies 690 using a debugger; using a map of defects on a storage device (e.g., a hard disk, memory card, etc.) to form internal test values to impede moving and/or copying HPE 655 to other electronic appliances 600; using kernel code that contains false branches and other complications in flow of control to disguise internal processes to some degree from disassembly or other efforts to discover details of processes; using 'self-generating' code (based on the output of a co-sine transform, for example) such that detailed and/or complete instruction sequences are not stored explicitly on storage devices and/or in active memory but rather are generated as needed; using code that 'shuffles' memory locations used for data values based on operational parameters to complicate efforts to manipulate such values; using any software and/or hardware memory management resources of electronic appliance 600 to 'protect' the operation of HPE 655 from other processes, functions, etc. Although such a software-based tamper resistant barrier 674 may provide a fair degree of security, it typically will not be as secure as the hardware-based tamper resistant barrier 502 provided (at least in part) by SPU 500." ('193 80:40-65; Fig. 10)</p> <p>12. "FIG. 12 also shows that ROS 602 may provide one or more SPEs 503 and/or one or more HPEs 655. As discussed above, HPE 655 may 'emulate' an SPU 500 device, and such HPEs 655 may be integrated in lieu of (or in addition to) physical SPUs 500 for systems that need higher throughput. Some security may be lost since HPEs 655 are typically protected by operating system security and may not provide truly secure processing. Thus, in the preferred embodiment, for high security applications at least, all secure processing should take place within SPE 503 having an execution space within a physical SPU 500 rather than a HPE 655 using software operating elsewhere in electronic appliance 600." ('193 88:31-43)</p> <p>13. "As discussed above in connection with FIG. 12, each electronic appliance 600 in the preferred embodiment includes one or more SPEs 503 and/or one or more HPEs 655. These secure processing environments each provide a protected execution space for performing tasks in a secure manner." ('193 104:39-44)</p> <p><b>Extrinsic:</b></p> <p>1. Host processor: "1. A processor that controls all or part of a user application network. 2. In a network, the processing unit in which resides the access method for the network. ... 4. A processing unit that executes the access method for attached communication controllers." (IBM)</p> <p>2. "Host Processing Environment (HPE): A software-only realization of the PPE, protected from tampering by appropriate software techniques. No longer preferred because of the potential confusion between the 'H' in the acronym and</p> |

|     | Claim<br>Term/Phrase                         | Evidence Supporting MS Construction   |
|-----|--|---|
|     |  | <p>'H' as in 'Hardware' (which this isn't). [REPLACEMENT UNCERTAIN]" (IT Glossary, "Obsolete Terminology Section,"<sup>2</sup> 3/7/95, IT00709621)</p> <p>3. "Secure Processing Environment (SPE): A hardware-supported realization of the PPE, protected from tampering by physical security techniques. No longer preferred because of the potential confusion between the 'S' in the acronym and 'S' as in 'Software' (which this isn't). [REPLACEMENT UNCERTAIN]" (IT Glossary, "Obsolete Terminology Section" 5/12/95, IT00028302)</p> <p>4. Environment: See InterTrust node: "A computer that is enabled for processing of DigiBox containers by installation of a PPE, which may be either hardware or software based. A node may include application software and/or operating system integration. The node is also termed the <i>environment</i>." (IT Glossary, 8/21/95, TD00068B, IT00032375)</p>   |
| 17. | <p>identifier</p> <p>193.15</p> <p>912.8</p> | <p><b>Intrinsic:</b></p> <p>1. "Portable appliance 2600 RAM 534 may contain, for example, information which can be used to uniquely identify each instance of the portable appliance. This information may be employed (e.g. as at least a portion of key or password information) in authentication, verification, decryption, and/or encryption processes." ('193 230:22-27)</p> <p>2. "Provide very flexible and extensible user identification according to individuals, installations, by groups such as classes, and by function and hierarchical identification employing a hierarchy of levels of client identification (for example, client organization ID, client department ID, client network ID, client project ID, and client employee ID, or any appropriate subset of the above)." ('193 25:31-38)</p> <p>3. "Fingerprinting is useful in providing an ability to identify who extracted information in clear form <i>[sic]</i> a VDE container, or who made a copy of a VDE object or a portion of its contents." ('193 37:27-31)</p> <p>4. "All load modules 1100 for use by SPE 503 are preferably referenced by a load module execution manager 568 that maintains and scans a list of available load modules and selects the appropriate load module for execution. If the load module is not present within SPE 503, the task is 'slept' and LMEM 568 may request that the load module 1100 be loaded from secondary storage 562. This request may be in the form of an RPC call to secure database manager 566 to retrieve the load module and associated data structures, and a call to encrypt/decrypt manager 556 to decrypt the load module before storing it in memory allocated by memory manager 578." ('193 111:47-58)</p> <p>5. "In somewhat more detail, the preferred embodiment executes a load module 1100 by passing the load module execution manager 568 the name (e.g., VDE ID) of the desired load module 1100. LMEM 568 first searches the list of 'in memory' and 'built-in' load modules 572. If it cannot find the desired load</p> |

<sup>2</sup> Some terms were "defined" in an "Obsolete Terminology Section" of certain IT Glossaries. This section was described in such documents as: "This section identifies terms that have been used in earlier documents to describe various VDE concepts, but that are, for various reasons, no longer preferred." (See, e.g., IT00028302)

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>module 1100 in the list, it requests a copy from the secure database 610 by issuing an RPC request that may be handled by ROS secure database manager 744 shown in FIG. 12.” (‘193 111:59-67)</p> <p>6. “For each VDE item loaded into SPE 503, Secure Database manager 566 in the preferred embodiment may search a master list for the VDE item ID, and then check the corresponding transaction tag against the one in the item to ensure that the item provided is the current item. Secure Database Manager 566 may maintain list of VDE item ID and transaction tags in a ‘hash structure’ that can be paged into SPE 503 to quickly locate the appropriate VDE item ID. In smaller systems, a look up table approach may be used. In either case, the list should be structured as a pagable <i>[sic]</i> structure that allows VDE item ID to be located quickly.” (‘193 124:8-18)</p> <p>7. “A stipulation that the traveling object may be used on certain one or more installations or installation classes or users or user classes where classes correspond to a specific subset of installations or users who are represented by a predefined class identifiers stored in a secure database 610.” (‘193 131:40-45)</p> <p>8. “A load module 1100 is able to perform its function only when executed in the protected environment of an SPE 503 or an HPE 655 because only then can it gain access to the protected elements (e.g., UDEs 1200, other load modules 1100) on which it operates. Initiation of load module execution in this environment is strictly controlled by a combination of access tags, validation tags, encryption keys, digital signatures and/or correlation tags. Thus, a load module 1100 may only be referenced if the caller knows its ID and asserts the shared secret correlation tag specific to that load module. The decrypting SPU may match the identification token and local access tag of a load module after decryption. These techniques make the physical replacement of any load module 1100 detectable at the next physical access of the load module.” (‘193 139:41-55)</p> <p>9. “These shared secrets may be used during communications processes to permit PPEs 650 to authenticate the identity of other PPEs and/or users.” (‘193 214:39-41)</p> <p>10. “As another example, interpreter 508 may provide application 506 with an element identification (e.g., a hexadecimal value or other identifier) that corresponds to the headline information within the newspaper style content (block 558). Application 506 may then ask electronic appliance 500 to provide it with the Headline (or other) content information 102 within container 100 by providing appropriate content information to electronic appliance 500 via APL 504 (block 560).” (‘861 12:63 - 13:4)</p> <p>11. “It is preferable that an extremely secure encryption/decryption technique be used as an aspect of authenticating the identity of electronic appliances 600 that are establishing a communication channel and securing any transferred permission, method, and administrative information.” (‘193 67:21-26)</p> <p>12. “As part of the initialization process, the PPE 650 may generate internally or the manufacturer may generate and supply, one or more pairs of site-specific public keys 2815 and private keys 2816. These are used by the PPE 650 to prove its identity.” (‘193 209:63-67)</p> |

|     | Claim<br>Term/Phrase   | Evidence Supporting MS Construction  |
|-----|--|--|
|     |  | <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Identifier: "1. One or more characters used to identify or name a data element and possibly to indicate certain properties of that data element. 2. In programming languages, a token that names a data object such as a variable, an array, a record, a subprogram or a function." (IBM)</li> <li>2. Identifier: "1. In computing, a character or group of characters used to identify, indicate or name a body of data. 2. In computing, a name or string of characters employed to identify a variable, procedure, data structure or some other element of a program." (Longley)</li> </ol>  |
| 18. | <p>protected processing environment</p> <p>683.2</p> <p>721.34</p> | <p>See also "secure"</p> <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Prosecution History of Application 08/778,256 (continuation of '891 Patent, issued as U.S. Patent No. 5,949,876), Amendment, 1/20/98, pp. 58-60: <ol style="list-style-type: none"> <li>a. "Independent claims 65 and 76 each recite a 'protected processing environment.' ... Griffeth et al. [U.S. Pat. No. 5,505,837], Yamamoto [U.S. Pat. No. 5,508,913] and Wyman [U.S. Pat. No. 5,260,999] do not disclose these aspects of these claims.</li> <li>b. The system disclosed in Griffeth et al is designed to allow negotiation to proceed in an environment in which a negotiating party does not disclose information about its negotiation goals to the other negotiating party. ... Griffeth et al. does not disclose any privacy protection mechanism and neither teaches nor suggests any secure processing environment or that any operations (e.g., integration or execution) occur securely. Indeed, Griffeth contains no suggestion that any protection mechanism is needed to maintain negotiation goals in privacy, since Griffeth does not suggest that the other party may try to improperly discover information which is intended to remain private.</li> <li>c. Yamamoto states the following: 'Here, the data is enciphered by the data encipher apparatuses 26 so as to maintain confidentiality.' Col. 3, lines 46-47. Since Yamamoto makes no other reference to the encipherment, or to the apparatuses 26, it is impossible to determine how the data encipherment is used, or the roles it plays in the disclosed apparatus. From an examination of Fig. 3, however, it appears that the data encipher apparatuses 26 are placed on connections between a particular site and other, physically separated sites. For example, customer office 23b is connected to sub-center 22 by a line, which apparently represents a communication path. That line connects directly to a data encipher apparatus 26 in customer office 23b, and to another data encipher apparatus 26 in sub-center 22.</li> <li>d. Thus, it appears that the data encipher apparatuses 26 are used, in some undisclosed manner, to encipher at least some data which travels among physically separated locations. It is possible to imagine, for example, that data is enciphered prior to being sent out on an insecure public transmission line, and is then deciphered once received in a new location.</li> <li>e. Yamamoto does not disclose, however, that the processing environments are themselves secure, or that either execution or integration occur in a secure manner or in a secure environment. Indeed, Yamamoto contains no suggestion that security within a processing environment would even be desirable. By</li> </ol> </li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>suggesting that data is deciphered once it enters an office (e.g., office 23b), in fact, Yamamoto teaches away from a secure environment, since it would appear that the data is used 'in the clear' within the office, with no suggested protection beyond a simple password for the computer.</p> <p>f. Wyman is equally deficient regarding these elements. Although Wyman specifies that a license may contain a digital signature, therefore rendering the license unforgeable (Col. 14, lines 24-54), Wyman neither teaches nor suggests that the processing environment is itself secure or that any operations occur in a secure manner. The Wyman digital signatures no more suggest a secure processing environment than the requirement that paper contracts be signed in ink suggests that the contracts will be created, read or negotiated in a secure location."</p> <p>2. "The role of go-between 4700 may, in some circumstances, be played by one of the participant's SPU's 500 (PPEs), since SPU (PPE) behavior is not under the user's control, but rather can be under the control of rules and controls provided by one or more other parties other than the user (although in many instances the user can contribute his or her own controls to operate in combination with controls contributed by other parties)." ('683 24:26-33)</p> <p>3. "SPU 500 provides a tamper-resistant protected processing environment ("PPE") in which processes and transactions can take place securely and in a trusted fashion." ('683 16:60-62)</p> <p>4. "The computer 3372 may then execute the operational materials 3472 from its hard disk 3376 to provide software-based protected processing environment 650 and associated software-based tamper resistant barrier 672)." ('900 231:27-31)</p> <p>5. "The special purpose secure circuitry provided by the present invention includes at least one of: a dedicated semiconductor arrangement known as a Secure Processing Unit (SPU) and/or a standard microprocessor, microcontroller, and/or other processing logic that accommodates the requirements of the present invention and functions as an SPU." ('193 20:58-63)</p> <p>6. "This means that a VDE SPU can employ (share) circuitry elements of a 'standard' CPU. For example, if a 'standard' processor can operate in protected mode and can execute VDE related instructions as a protected activity, then such an embodiment may provide sufficient hardware security for a variety of applications and the expense of a special purpose processor might be avoided." ('193 21:11-17)</p> <p>7. "Different protected processing environments (secure execution spaces) might examine different subsets of the multiple digital signatures--so that compromising one protected processing environment (secure execution space) will not compromise all of them." ('721 7:19-23)</p> <p>8. "The assurance level III appliance 61C shown is a general purpose personal computer equipped with a hardware-based secure processing unit 132 providing and completely containing protected processing environment 108 (see Ginter et al. FIGS. 6 and 9 for example). A silicon-based special purpose integrated circuit security chip is relatively more tamper-resistant than implementations relying on software techniques for some or all of their tamper-resistance." ('721 16:64 - 17:5)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>9. "FIG. 10 is a block diagram of one example of a software structure/architecture for Rights Operating System ('ROS') 602 provided by the preferred embodiment. In this example, ROS 602 includes an operating system ('OS') 'core' 679, a user Application Program Interface ('API') 682, a 'redirector' 684, an 'intercept' 692, a User Notification/Exception Interface 686, and a file system 687. ROS 602 in this example also includes one or more Host Event Processing Environments ('HPEs') 655 and/or one or more Secure Event Processing Environments ('SPEs') 503 (these environments may be generically referred to as 'Protected Processing Environments' 650). HPE(s) 655 and SPE(s) 503 are self-contained computing and processing environments that may include their own operating system kernel 688 including code and data processing resources." ('193 79:36-39)</p> <p>10. "A given electronic appliance 600 may include any number of SPE(s) 503 and/or any number of HPE(s) 655. HPE(s) 655 and SPE(s) 503 may process information in a secure way, and provide secure processing support for ROS 602. For example, they may each perform secure processing based on one or more VDE component assemblies 690, and they may each offer secure processing services to OS kernel 680. In the preferred embodiment, SPE 503 is a secure processing environment provided at least in part by an SPU 500. Thus, SPU 500 provides the hardware tamper-resistant barrier 503 surrounding SPE 503. SPE 503 provided by the preferred embodiment is preferably: small and compact[,] loadable into resource constrained environments such as for example minimally configured SPUs 500[,] dynamically updatable[,] extensible by authorized users[,] integratable into object or procedural environments[, and] secure." ('193 79:39-59)</p> <p>11. "As shown in FIG. 13, SPE 503 (PPE 650) includes the following service managers/major functional blocks in the preferred embodiment:<br/> Kernel/Dispatcher 552<br/> Channel Services Manager 562<br/> SPE RPC Manager 550<br/> Time Base Manager 554<br/> Encryption/Decryption Manager 556<br/> Key and Tag Manager 558<br/> Summary Services Manager 560<br/> Authentication Manager/Service Communications Manager 564<br/> Random Value Generator 565<br/> Secure Database Manager 566<br/> Other Services 592.<br/> Each of the major functional blocks of PPE 650 is discussed in detail below."<br/> ('193 105:23-41)</p> <p>12. "I. SPE Kernel/Dispatcher: 552The Kernel/Dispatcher 552 provides an operating system 'kernel' that runs on and manages the hardware resources of SPU 500. This operating system 'kernel' 552 provides a self-contained operating system for SPU 500; it is also a part of overall ROS 602 (which may include multiple OS kernels, including one for each SPE and HPE ROS is controlling/managing). Kernel/dispatcher 552 provides SPU task and memory management, supports</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>internal SPU hardware interrupts, provides certain 'low level services,' manages 'DTD' data structures, and manages the SPU bus interface unit 530. Kernel/dispatcher 552 also includes a load module execution manager 568 that can load programs into secure execution space for execution by SPU 500." ('193 105:43-57) (see also Fig. 13)</p> <p>13. "In addition, memory management provided by memory manager 578 operating at least in part based on hardware-based MMU 540 may securely implement and enforce a memory architecture providing multiple protection domains. In such an architecture, memory is divided into a plurality of domains that are largely isolated from each other and share only specific memory areas under the control of the memory manager 578. An executing process cannot access memory outside its domain and can only communicate with other processes through services provided by and mediated by privileged kernel/dispatcher software 552 within the SPU 500. Such an architecture is more secure if it is enforced at least in part by hardware within MMU 540 that cannot be modified by any software-based process executing within SPU 500." ('193 109:46-60)</p> <p>14. "Secure VDE hardware (also know as SPUs for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host Processing Environments (HPEs)), operate in conjunction with secure communications, system integration software, and distributed software control information and support structures, to achieve the electronic contract/rights protection environment of the present invention. Together, these VDE components comprise a secure, virtual, distributed content and/or appliance control, auditing (and other administration), reporting, and payment environment. In some embodiments and where commercially acceptable, certain VDE participants, such as clearinghouses that normally maintain sufficiently physically secure non-VDE processing environments, may be allowed to employ HPEs rather VDE hardware elements and interoperate, for example, with VDE end-users and content providers." ('193 13:7-23)</p> <p>15. "Each PPE 650 needs to be initialized before it can be used. Initialization may occur at the manufacture site, after the PPE 650 has been placed out in the field, or both. The manufacturing process for PPE 650 typically involves embedding within the PPE sufficient software that will allow the device to be more completely initialized at a later time. This manufacturing process may include, for example, testing the bootstrap loader and challenge-response software permanently stored within PPE 650, and loading the PPE's unique ID. These steps provide a basic VDE-capable PPE 650 that may be further initialized (e.g., after it has been installed within an electronic appliance 600 and placed in the field). In some cases, the manufacturing and further initialization process may be combined to produce 'VDE ready' PPEs 650." ('193 223:30-44)</p> <p>16. "In one example, a person with a laptop 5102 or other computer lacking a PPE 650 wishes nonetheless to take advantage of a subset of secure item delivery services." ('683 62:17-20)</p> <p>17. "Claims 7-11, ... 99-111 ... are rejected under 35 U.S.C. 103(a) as being unpatentable over Fischer (5,412,717) in view of Narasimhalu et al (5,499,298). Fischer discloses a method and apparatus including a system monitor which</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>limits the ability of a program about to be executed to the use of predefined resources, .... The set of authorities and restrictions are referred to as 'program authorization information' or 'PAI'. ... A comparison of independent claim 7 to Fischer to derive the similarities and differences between the claimed invention and the prior art follows. ... memory containing a first rule corresponds to a first PAI under a first PCB... Here, Fischer provides a secure container in the form of a program, i.e. a governed item, having an associated PAI, i.e. at least one rule associated with the secure container. A protected processing environment ('PPE') protecting at least some information contained in the PPE, see Fischer Terminal A, and including hardware and/or software used for applying said first rule and the secure container in combination to at least in part govern at least one aspect of access to or use of the governed item, see Fischer at Figure 5 and column 10, lines 8-39 where the first rule in memory is first PCB providing a first PAI and the secure container is a program associated with a second PCB providing a first PAI and the secure container is a program associated with a second PCB having a second PAI associated with the governed item, i.e. the program. ... The difference between claim 7 and Fischer is that the PPE disclosed in Fischer is not explicitly disclosed as protected from tampering by a user of the first apparatus, i.e. terminal A. The Narasimhalu patent ... teaches a method and apparatus for controlling the dissemination of digital information [and] that the end user accesses the digital information with a tamper-proof controlled information access device." (Prosecution History for the 09/221,479 Patent Application, (issued as the '683), Office Action, 11/12/99, pp. 3-5 (TT00065799-801))</p> <p>18. "With respect to the remaining issues, Applicants respectfully disagree. For example, the Examiner objects to the use of 'environment' as indefinite and unclear. This word, however, is not used in isolation, but rather in the context of several longer phrases, all of which are defined in the specification. The phrase 'protected processing environment,' for example, is used in Claims 11 and 15-18 and described on at least, for example, pages 7-8 and 25 of the specification. The term 'virtual distribution environment' used in Claim 11 is described, for example, on page 7 of the specification. The terms are also described in the commonly copending application Serial Number 08/388,107 of Ginter et al., filed 13 February 1995, entitled 'System and Methods for Secure Transaction Management and Electronic Rights Protection.' A copy of the incorporated Ginter application can be provided to the Examiner upon request." (Prosecution History for the 08/689,754 Patent Application (issued as the '721), Amendment, 4/14/99, p. 13) (pp. 7, 7-8 and 25 of the original specification are '721 2:62 - 3:13, 2:62 - 3:34 and 8:6-28 of the issued patent)</p> <p>19. "Another approach to supporting COTS software would use the VDE software running on the user's electronic appliance to create one or more 'virtual machine' environments in which COTS operating system and application programs may run, but from which no information may be permanently stored or otherwise transmitted except under control of VDE." ('193 279:26-40)</p> <p>20. "VDE may be combined with, or integrated into, many separate computers and/or other electronic appliances. These appliances typically include a secure</p> |



| Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|----------------------|--|
|                      | <p>subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred embodiment comprises one or more 'protected processing environments' ...." ('193 9:22-29)</p> <p>21. "The operating system 602 may also support at least one 'application' 608. Generally, 'application' 608 is hardware and/or software specific to the context of appliance 600. For example, if appliance 600 is a personal computer, then 'application' 608 could be a program loaded by the user, for instance, a word processor, a communications system or a sound recorder. If appliance 600 is a television controller box, then application 608 might be hardware or software that allows a user to order videos on demand and perform other functions such as fast forward and rewind. In this example, operating system 602 provides a standardized, well defined, generalized 'interface' that could support and work with many different 'applications' 608." ('193 60:51-64)</p> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. Processing: "1. The performance of logical operations and calculations on data, including temporary retention of data in processor storage while the data is being operated on." (IBM)</li> <li>2. Environment: "1. The aggregate of external circumstances, conditions, and objects that affect the development, operation, and maintenance of a system. 2. In computer security, those factors, both internal and external, of an ADP system that help to define the risks associated with its operation." (Longley)</li> <li>3. "The InterTrust architecture employs three principal components: ... The InterRights Point software provides 'Protected Processing Environment™' technology for manipulating information in DigiBox containers and for securely implementing business rules." (Panel: The InterTrust Commerce Architecture, D. Van Wie et al., 20<sup>th</sup> NISSC, p. 2, 1997)</li> <li>4. Environment: See InterTrust node: "A computer that is enabled for processing of DigiBox containers by installation of a PPE, which may be either hardware or software based. A node may include application software and/or operating system integration. The node is also termed the <i>environment</i>." (IT Glossary, 8/21/95, TD00068B, IT00032375)</li> <li>5. Protected Processing Environment (PPE) technology: "The InterTrust technology that provides the protected software environment within the InterRights Point. Protected Processing Environment technology is responsible for the encryption/decryption of data, protected processing of DigiBox containers, and other secure operations, such as protected database access." (IT Glossary, 1997-1998, ML00012B)</li> <li>6. Protected Processing Environment (PPE): "The PPE is the secure part of a VDE node: either a hardware or software-protected environment in which VDE mechanisms run without external interference. There are various PPE realizations (e.g., physically protected hardware) appropriate to different operational requirements" (IT Glossary, 3/7/95, IT00709619)</li> <li>7. Secure Processing Unit: "The physically secure hardware component of the SPE: a processor with local memory and non-volatile storage. The SPE consists of the</li> </ol> |

|     | Claim Term/Phrase   | Evidence Supporting MS Construction   |
|-----|---|---|
|     |   | <p>SPU itself and the SPE software running on the SPU.” (IT Glossary, 3/7/95, IT00709620)</p> <p>8. Protected Processing Environment (PPE): “An InterTrust <i>node</i> has a unique <i>node ID</i> and contains a <i>Protected Processing Environment (PPE)</i> which performs <i>operations on containers and control structures</i> under rules specified by <i>PERCs</i> and which may be realized in a tamper resistant hardware component or in tamper-resistant software and a <i>protected database</i>, which stores <i>control objects</i> and <i>InterTrust applications</i>, operating outside the <i>PPE</i>, which manipulate <i>content</i> and <i>control objects</i> through requests to the <i>PPE</i>” (IT Glossary, 4/6/95, IT00028206)</p> <p>9. “All the terms in italics have specific definitions (in the glossary) with respect to InterTrust.”</p> <p>10. “Global replace of ‘VDE’ with ‘InterTrust’ to match new terminology.” (IT Glossary, 4/6/95, IT00028206)</p> <p>11. Protected Environment: “A portion of the node software that uses, and protects, the protected node data such as cryptographic keys. The protected environment is responsible for performing all the protected functions for manipulating containers and content; that is, all the operations governed by controls.” (IT Glossary, 5/12/95, IT00028294)</p> <p>12. Protected Processing Environment: (alternate definition): “The protected environment in which the cryptographic and control functions of InterTrust run. The PPE may be protected environmentally (e.g., as a physically protected server machine) or may employ software-based tamper resistance techniques.” (IT Glossary, 8/21/95, TD00068B, IT00032377)</p> <p>13. Secure Processing Environment (SPE): “A hardware-supported realization of the PPE, protected from tampering by physical security techniques. No longer preferred because of the potential confusion between the ‘S’ in the acronym and ‘S’ as in ‘Software’ (which this isn’t).. [REPLACEMENT UNCERTAIN]” (IT Glossary, “Obsolete Terminology Section,” 5/12/95, IT00028302)</p> <p>14. Protected Processing Environment (PPE): “The InterTrust protected software environment within the InterTrust Commerce Node. The PPE is responsible for the encryption/decryption of data, protected processing of DigiBox containers, and other secure operations, such as database access.” (IT Glossary, 11/17/96, TD00189J, IT00035871)</p> <p>15. Process: “(1) in computing, the active system entity through which programs run. The entity in a computer system to which authorizations are granted; thus the unit of accountability in a computer system. (2). In computing, a program in execution. ... (4) In computing, a program is a static piece of code and a process is the execution of that code.” (Longley)</p> |
| 19. | <p>secure, securely</p> <p>193.1, 193.11, 193.15</p> <p>683.2</p> <p>721.34</p> | <p><b>Intrinsic:</b></p> <p>Because this term is indefinite and used inconsistently, each use of “secure” and forms thereof in the asserted patents is relevant and herein included by reference. The following examples are illustrative.</p> <p>1. “HPEs 655 may be provided in two types: secure and not secure.” (‘193 80:8-9)</p> <p>2. “Because secondary storage 652 is not secure, SPE 503 must encrypt and cryptographically seal (e.g., using a one-way hash function initialized with a</p>  |

|  | Claim<br>Term/Phrase             | Evidence Supporting MS Construction   |
|--|----------------------------------|---|
|  | 861.58<br>891.1<br>912.8, 912.35 | <p>secret value known only inside the SPU 500) each swap block before it writes it to secondary storage.” (‘193 107:39-42)</p> <p>3. “Insecure external memory may reduce the wait time for swapped pages to be loaded into SPU 500, but will still incur substantial encryption/decryption penalty for each page.” (‘193 125:56-59)</p> <p>4. “The following is a non-exhaustive list of some of the advantageous features provided by ROS 602 in the preferred embodiment:</p> <p>...</p> <p>Secure</p> <p>secure communications</p> <p>secure control functions</p> <p>secure virtual memory management</p> <p>information control structures protected from exposure</p> <p>data elements are validated, correlated and access controlled</p> <p>components are encrypted and validated independently</p> <p>components are tightly correlated to prevent unauthorized use of elements</p> <p>control structures and secured executables are validated prior to use to protect against tampering</p> <p>integrates security considerations at the I/O level</p> <p>provides on-the-fly decryption of information at release time</p> <p>enables a secure commercial transaction network</p> <p>flexible key management features” (‘193 72:52 - 73:38)</p> <p>5. “ROS 602 generates component assemblies 690 in a secure matter. As shown graphically in FIGS. 11I and 11J, the different elements comprising a component assembly 690 may be ‘interlocking’ in the sense that they can only go together in ways that are intended by the VDE participants who created the elements and/or specified the component assemblies. ROS 602 includes security protections that can prevent an unauthorized person from modifying elements, and also prevent an unauthorized person from substituting elements.” (‘193 84:60 - 85:2)</p> <p>6. “Because of VDE security, including use of effective encryption, authentication, digital signature, and secure database structures, the records contain within a VDE card arrangement may be accepted as valid transaction records for government and/or corporate recordkeeping requirements.” (‘193 41:37-42)</p> <p>7. “In order to maintain security, SPE 503 must encrypt and cryptographically seal each block being swapped out to a storage device external to a supporting SPU 500, and must similarly decrypt, verify the cryptographic seal for, and validate each block as it swapped into SPU 500.” (‘193 125:60-64)</p> <p>8. “As mentioned above, memory external to SPU 500 may not be secure. Therefore, when security is required, SPU 500 must encrypt secure information before writing it to external memory before using it.” (‘193 71:32-36)</p> <p>9. “Only those processes that execute completely within SPEs 503 (and in some cases, HPEs 655) may be considered to be truly secure. Memory and other resources external to SPE 503 and HPEs 655 used to store and/or process code and/or data to be used in secure processes should only receive and handle that information in encrypted form unless SPE 503/HPE 655 can protect secure process code and/or data from non-secure processes.” (‘193 81:12-19)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>10. "From time to time, two parties (e.g., PPEs A and B), will need to establish a communication channel that is know by both parties to be secure form eavesdropping, secure from tampering, and to be in use solely by the two parties whose identifies are correctly known to each other." ('193 218:33-37)</p> <p>11. "Since all secure communications are at least in part encrypted and the processing inside the secure subsystem is concealed form outside observation and interference, the present invention ensures that content control information can be enforced." ('193 46:4-8)</p> <p>12. "VDE 100 provided by the preferred embodiment has sufficient security to help ensure that it cannot be compromised short of a successful 'brute force attack,' and so that the time and cost to succeed in such a 'brute force attack' substantially exceeds any value to be derived. In addition, the security provided by VDE 100 compartmentalizes the internal workings of VDE so that a successful 'brute force attack' would compromise only a strictly bounded subset of protected information, not the entire system." ('193 199:38-47)</p> <p>13. "Integrity of VDE Security: There are many ways in which a PPE 650 might be compromised. The goal of the security provided by VDE 100 is to reduce the possibility that the system will be compromised, and minimize the adverse effects if it is compromised. The basic cryptographic algorithm that are used to implement VDE 100 are assumed to be safe (cryptographically strong). These include the secret-key encryption of content, public-key signatures for integrity verification, public-key encryption for privacy between PPEs 650 or between a PPE and a VDE administrator, etc. Direct attack on these algorithms is assumed to be beyond the capabilities of an attacker. For domestic versions of VDE 100 some of this probably a safe assumption since the basic building blocks for control information have sufficiently long keys and are sufficiently proven. The following risks of threat or attacks may be significant: Unauthorized creation or modification of component assemblies (e.g., budgets); Unauthorized bulk disclosure of content; Compromise of one or more keys" ('193 221:1-21)</p> <p>14. See also prior art referenced in the relevant file histories, e.g., Stefik; Tygar et al., "Dyad: A System for Using Physically Secure Coprocessors," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (May 1991).</p> <p>15. "VDE can: (a) audit and analyze the use of content, (b) ensure that content is used only in authorized ways, and (c) allow information regarding content usage to be used only in ways approved by content users." ('193 4:51-56)</p> <p>16. "Even if the object is stored locally to the VDE node, it may be stored as a secure or protected object so that it is not directly accessible to a calling process." ('193 192:14-17)</p> <p>17. "An attacker would gain little benefit from intercepting this information since it is transmitted in protected form; she would have to compromise electronic appliance 600(1) or 600(N) (or the SPU 500(1), 500(N)) in order to access this information in unprotected form." ('193 228:25-30)</p> <p>18. "VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties." ('193 6:33-35)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>19. "It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g, summary, table of contents) and secured content control information which ensures the performance of control information." ('193 15:41-46)</p> <p>20. "Because of the breadth of issues resolved by the present invention, it can provide the emerging 'electronic highway' with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions." ('193 17:22-28)</p> <p>21. "VDE can satisfy the requirements of widely differing electronic commerce and data security applications by, in part, employing this general purpose transaction management foundation to securely process VDE transaction related control methods." ('193 25:52-57)</p> <p>22. "HPE(s) and SPE(s) ... may each perform secure processing based on one or more VDE component assemblies 690, and they may each offer secure processing services to OS kernel 680." ('193 79:41-46)</p> <p>23. "VDE methods 1000 are designed to provide a very flexible and highly modular approach to secure processing." ('193 181:18-19)</p> <p>24. "In these cases, secure processing steps performed by an SPU typically must be segmented into small, securely packaged elements that may be 'paged in' and 'paged out' of the limited available internal memory space." (69:43-47)</p> <p>25. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... VDE employs special purpose hardware distributed throughout some or all locations of a VDE implementation: a) said hardware controlling important elements of: content preparation (such as causing such content to be placed in a VDE content container and associating content control information with said content), content and/or electronic appliance usage auditing, content usage analysis, as well as content usage control; and b) said hardware having been designed to securely handle processing load module control activities, wherein said control processing activities may involve a sequence of required control factors" ('193 21:43 - 22:31)</p> <p>26. "Memory manager 578 and virtual memory manager 580 in the preferred embodiment manage ROM 532 and RAM 534 memory within SPU 500 in the preferred embodiment. Virtual memory manager 580 provides a fully 'virtual' memory system to increase the amount of 'virtual' RAM available in the SPE secure execution space beyond the amount of physical RAM 534a provided by SPU 500. Memory manager 578 manages the memory in the secure execution space, controlling how it is accessed, allocated and deallocated. SPU MMU 540, if present, supports virtual memory manager 580 and memory manager 578 in the preferred embodiment. In some 'minimal' configurations of SPU 500 there may be no virtual memory capability and all memory management functions will be handled by memory manager 578. Memory management can also be used to help enforce the security provided by SPE 503. In some classes of SPUs 500, for example, the kernel memory manager 578 may use hardware memory management unit (MMU) 540 to provide page level protection within the SPU 500. Such a hardware-based memory management system provides an effective</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>mechanism for protecting VDE component assemblies 690 from compromise by 'rogue' load modules." ('193 109:24-45)</p> <p>27. "When a method core 1000' references a load module 1100, a load module is loaded into the SPE 503, decrypted, and then either passed to the electronic appliance microprocessor for executing in an HPE 655 (if that is where it executes), or kept in the SPE (if that is where it executes)." ('193 139:28-31)</p> <p>28. "The role of go-between 4700 may, in some circumstances, be played by one of the participant's SPU's 500 (PPEs), since SPU (PPE) behavior is not under the user's control, but rather can be under the control of rules and controls provided by one or more other parties other than the user (although in many instances the user can contribute his or her own controls to operate in combination with controls contributed by other parties)." ('683 24:26-33)</p> <p>29. "Load modules are not necessarily directly governed by PERCs 808 that control them, nor must they contain any time/date information or expiration dates. The only control consideration is the preferred embodiment is that one or more methods 1000 reference them using a correlation tag (the value of a protected object created by the load module's owner, distributed to authorized parties for inclusion in their methods, and to which access and use is controlled by one or more PERCs 808). If a method core 1000' references a load module 1100 and asserts the proper correlation tag (and the load module satisfies the internal tamper checks for the SPE 503), then the load module can be loaded and executed, or it can be acquired from, shipped to, updated, or deleted by, other systems." ('193 139:60 - 140:6)</p> <p>30. "ROS 602 also provides a tagging and sequencing scheme that may be used within loadable component assemblies 690 to detect tampering by substitution. Each element comprising a component assembly 690 may be loaded into a SPU 500, decrypted using encrypt/decrypt engine 522, and then tested/compared to ensure that the proper element has been loaded. ...In addition, a validation/correlation tag stored under the encrypted layer of the loadable element may be compared to make sure it matches on or more tags provided by a requesting process. This prevents unauthorized use of information. As a third protection, a device assigned tag (e.g., a sequence number) stored under an encryption layer of loadable element may be checked to make sure it matches a corresponding tag value expected by SPU 500. This prevents substitution of older elements. Validation/correlation tags are typically passed only in secure wrappers to prevent plaintext exposure of this information outside of SPU 500." ('193 87:41-62)</p> <p>31. "Key and Tag Manager 558 also provides service relating to tag generation and management. In the preferred embodiment, transaction and access tags are preferably stored by SPE 503 (HPE 665) in protected memory (e.g., within the NVRAM 534b of SPU 500). These tags may be generated by key and tag manager 558. They are used to, for example, check access rights to, validate and correlate data elements. For example, they may be used to ensure components of the secured data structures are not tampered with outside of the SPU 500." ('193 120:59 - 121:1)</p> <p>32. "Initiation of load module execution in this environment is strictly controlled by a</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>combination of access tags, validation tags, encryption keys, digital signatures, and/or correlation tags. Thus, a load module 1100 may only be referenced if the caller knows its ID and asserts the shared secret correlation tag specific to that load module. The decrypting SPU may match the identification token and a local access tag of a load module after decryption. These techniques make the physical replacement of any load module 1100 detectable at the next physical access of a load module." ('193 139:45-55)</p> <p>33. "Meters and budgets are common examples of this. Expiration dates cannot be used effectively to prevent substitution of the previous copy of a budget UDE 1200. To secure these frequently updated items, a transaction tag is generated and included in the encrypted item each time that item is updated. A list of all UDE items IDs and the current transaction tags for each item is maintained as part of the secure database 610." ('193 143:13-20)</p> <p>34. "UDEs 1200 are preferably encrypted using a site specific key once they are loaded into a site. This site-specific key marks a validation tag that may be derived from a cryptographically strong pseudo-random sequence by the SPE 503 and updated each time the record is written back to the secure database 610. This technique provided reasonable assurance that the UDE 1200 has not been tampered with nor submitted when it is requested by the system for the next use." ('193 143: 29-37)</p> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "No data system can be made secure without physical protection of some part of the equipment." (Davies, p. 3)</li> <li>2. "Security is a negative attribute. We judge a system to be secure if we have not been able to design a method of misusing it which gives some advantage to the attacker." (Davies, p. 4)</li> <li>3. "Various criteria exist for secure systems - U.S. Dept. of Defense Trusted Computer Security Evaluation Criteria (TCSEC), the Orange Book, Red Book, European and Canadian guidelines, U.S. National Institute of Standards and Technology, and United Kingdom guidelines." (Neumann, p. 233)</li> <li>4. Security: "1. Protection against unwanted behavior. In present usage, computer security includes properties such as confidentiality, integrity, availability, prevention of denial of service and prevention of generalized misuse. 2. The property that a particular security policy is enforced, with some degree of assurance. 3. Security is sometimes used in the restricted sense of confidentiality, particularly in the case of multilevel security (that is, multilevel confidentiality)." Multilevel Security: "A confidentiality policy based on the relative ordering of multilevel security labels (really multilevel confidentiality, ex. - no adverse flow of information with respect to sensitivity of information)" (Neumann, Glossary and p. 225)</li> <li>5. "There are two principal objectives: secrecy (or privacy), to prevent unauthorized disclosure of data; and authenticity or integrity) [sic], to prevent the unauthorized modification of data.... Note, however, that whereas it can be used to detect message modification, it cannot prevent it. Encryption alone does not protect against replay, because an opponent could simply replay previous ciphertext."</li> </ol> |

| Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|----------------------|--|
|                      | <p>(Denning, p. 5)</p> <ol style="list-style-type: none"> <li>6. "A cipher is unconditionally secure if, no matter how much ciphertext is intercepted, there is not enough information in the ciphertext to determine the plaintext uniquely." (Denning, p. 5) (Davies, pp. 41, 380)</li> <li>7. "A cipher is computationally secure, or strong, if it cannot be broken by systematic analysis with available resources." (Denning, p. 5) (Davies, pp. 41, 370)</li> <li>8. Security: "The combination of integrity and secrecy, applied to data." (IT Glossary, 5/12/95, IT00028295)</li> <li>9. Secrecy: "The inability to obtain any information from data." (IT Glossary, 5/12/95, IT00028294)</li> <li>10. "... security includes concealment, integrity of messages, authentication of one communicating party by the other..." (Neumann, p. 8)</li> <li>11. "Computer security rests on confidentiality, integrity, and availability. The interpretations of these three aspects vary, as do the contexts in which they arise. Confidentiality is the concealment of information or resources. ... Confidentiality also applies to the existence of data, which is sometimes more revealing than the data itself. ... All mechanisms that enforce confidentiality require supporting services from the system. The assumption is that the security services can rely on the kernel, and other agents, to supply correct data. Thus, assumptions and trust underlie the confidentiality mechanisms. Integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change. Integrity includes data integrity (the content of the information) and origin integrity (the source of the data, often called authentication). Integrity mechanisms fall into two classes: prevention mechanisms and detection mechanisms. Protection mechanisms seek to maintain the integrity of the data by blocking any unauthorized attempts to change the data or any attempts to change the data in unauthorized ways. Detection mechanisms do not try to prevent violations of integrity; they simply report that the data's integrity is no longer trustworthy." (Bishop, pp. 4-6)</li> <li>12. "Definition 4-1. A security policy is a statement that partitions the states of the system into a set of authorized, or secure, states and a set of unauthorized, or nonsecure, states.<br/>A secure system is a system that starts in an authorized state and cannot enter an unauthorized state." (Bishop, p. 95)</li> <li>13. "24.5.1 Secure Systems Systems designed with security in mind have auditing mechanisms integrated with the system design and implementation." (Bishop, p. 706)</li> <li>14. "Computer security is assuring the secrecy, integrity, and availability of components of computing systems. The three principal pieces of a computing system subject attacks are hardware, software, and data. These three pieces, and the communications between them, constitute the basis of computer security vulnerabilities. This chapter has identified four kinds of attacks on computing systems: interruptions, interceptions, modifications, and fabrications. Three principles affect the direction of work in computer security. By the principle of easiest penetration, a computing system penetrator will use whatever means of</li> </ol> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>attack is the easiest; therefore. All aspects of computing system security need to be considered at once. By principle of timeliness, a system needs to be protected against penetration only long enough so that penetration is of no value to the penetrator. The principle of effectiveness states that controls must be usable and used in order to serve purpose. Controls can be applied at the levels of data, programs, the system, physical devices, communications links, the environment, and personnel. Sometimes several controls are needed to cover a single vulnerability, and sometimes one control addresses several problems at once.” (Pfleeeger, p. 4)</p> <p>15. See also InterTrust’s Rule 30(b)(6) testimony</p> <p>16. See also Microsoft PLR 4-2 Exhs. E &amp; F as revised, e.g.<br/> <u>Webster’s</u> (1947), p. 1540-41;<br/> <u>Pfleeeger</u>, p. 4-5;<br/> <u>Spencer, Personal Computer Dictionary</u>, p. 156;<br/> <u>The Computer Glossary</u>, p. 460;<br/> <u>McGraw-Hill Dictionary of Scientific and Technical Terms</u>, p. 1788;<br/> <u>Practical Unix Security</u> (O’Reilly 1991), p. 11-12;<br/> <u>Bishop, Computer Security</u> (2002) p. 3-24, 47;<br/> <u>Hoffman, Modern Methods for Computer Security and Privacy</u>, p. 134-35;<br/> <u>Mullender, ed., Distributed Systems</u> (Addison Wesley 2<sup>nd</sup> ed.), p. 367, 420;<br/> <u>Landwehr, “Formal Models for Computer Security”</u> (ACM 1981);<br/> <u>Merkle, “Protocols for Public Key Cryptosystems”</u> (IEEE 1980);<br/> <u>Cooper, Computer &amp; Communication Security</u>, p. 383;<br/> <u>Baker, The Computer Security Handbook</u>, p. 273;<br/> <u>Computer Security Handbook</u>, p. 389;<br/> <u>Matheson et al., Robustness and Security of Digital Watermarks</u>;<br/> <u>National Information Systems Security (INFOSEC) Glossary</u>, p. 49-50;<br/> <u>Internet Security Glossary</u> (RFC2828);<br/> <u>Tanenbaum, Modern Operating Systems</u> (1992), p. 181-82;<br/> IN64706-45, IN176319-72, IT735936 (integrity), IT735938-9<br/> IN00862862, IT1678-96, IT39208-26, IT702969-83, IT399877-80</p> <p>17. “Secure. Pertaining to the control of who can use an object and to the extent to which the object can be used by controlling the authority given to the user.”;<br/> “Computer Security. 1. Concepts, techniques, technical measures, and administrative measures used to protect the hardware, software and data of an information processing system from deliberate or inadvertent unauthorized acquisition, damage, destruction, disclosure, manipulation, modification or use or loss. 2. Protection resulting from the application of computer security.” (IBM)</p> <p>18. “Security: Freedom from risk or danger. Safety and assurance of safety”; “secure state - a condition in which none of the subjects in a system can access objects in an unauthorized manner...” (Russell, pp. 8-11, 113, 227, 420)</p> <p>19. “The protection of computer hardware and software from accidental or malicious access, use, modification, destruction, or disclosure.” (Booth)</p> <p>20. “Prevention of or protection against (a) access to information by unauthorized recipients or (b) intentional but unauthorized destruction or alteration of that information.” (Dictionary of Computing, p. 406)</p> |

|     | Claim Term/Phrase  | Evidence Supporting MS Construction  |
|-----|--|--|
|     |  | <p>21. "The quality or state of being cost-effectively protected from undue losses (e.g. loss of goodwill, monetary loss, loss of ability to continue operations, etc.)" (Longley).</p> <p>22. Hoffman, <u>Modern Methods for Computer Security &amp; Privacy</u>, p. 134</p> <p>23. "Protected Location: A memory location that can only be accessed by an authorized user or process."; "Protected domain: A set of access privileges to protected resources." (Dictionary of Computing)</p> <p>24. Protect: "To prevent unauthorized access to programs or a computer system; to shield against harm." (Webster's)</p> <p>25. Protection: "(1) (computing systems). See: Storage protection (2) (software). An arrangement for restricting access to or use of a all, or part, of a computer system."; Storage protection: "An arrangement for preventing access to storage for either reading or writing, or both." (Booth)</p> <p>26. IN00862862</p> <p>27. Security: "The combination of integrity and secrecy, applied to data." (IT Glossary, 5/12/95, IT00028295)</p> <p>28. "Secrecy: The inability to obtain any information from data." (IT Glossary, 5/12/95, IT00028294)</p> <p>29. Processing: "1. The performance of logical operations and calculations on datum including temporary retention of data in processor storage while the data is being operated on." (IBM)</p> <p>30. Process: "(1) in computing, the active system entity through which programs run. The entity in a computer system to which authorizations are granted; thus the unit of accountability in a computer system. (2) In computing, a program in execution... (4) In computing, a program is a static piece of code and a process is the execution of that code." (Longley)</p> <p>31. Processing: "In legislation, as defined by the U.K. Data Protection Act of 1984, pertaining to the amending, augmenting, deleting, or re-arranging of the data or extracting the information constituting the data and, in the case of personal data, processing means performing any of the abovementioned operations by reference to the data subject." (Longley)</p> |
| 20. | <p>secure container</p> <p>683.2</p> <p>861.58</p> <p>912.35</p> | <p><b>Intrinsic:</b></p> <p>1. "Anderson [U.S. Patent No. 5,537,526] does not explicitly address a secure container <i>per se</i>, but does place documents into containers [Fig. 8 202] and place restriction via links attached to documents ... which can include restrictions ... Such security tools are rightfully attached to a structure encapsulating the document, e.g. its container." (Prosecution History for the 08/805,804 Patent Application (issued as the '861), Office Action, 6/25/98, p. 5 (MSI 27417-25))</p> <p>2. "Claims 7-11, ... are rejected under 35 U.S.C. 103(a) as being unpatentable over Fischer (5,412,717) in view of Narasimhalu et al (5,499,298). ... The set of authorities and restrictions are referred to as 'program authorization information' or 'PAI'. ... A comparison of independent claim 7 to Fischer to derive the similarities and differences between the claimed invention and the prior art follows. ... Here, Fischer provides a secure container in the form of a program, i.e. a governed item, having an associated PAI, i.e. at least one rule associated with the secure container." (Prosecution History for the 09/221,479 Patent</p>  |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>Application (issued as the '683), Office Action, 11/12/99, pp. 3-4 (IT00065799-800))</p> <p>3. "1. (Amended) A rights management method comprising: (a) receiving an information signal; (b) steganographically decoding the received information signal to recover digital rights management control information <u>packaged within at least one secure digital container</u>; and (c) performing at least one rights management operation based at least in part on the recovered digital rights management control information. ...</p> <p>Remarks ... For example, amended Claims 1, 15 and 22 each recite a digital secure container in combination. Neither Rhoads [U.S. Patent No. 5,636,292], nor any of the other applied references, teaches or suggests the recited combination of features including any digital secure container." (Prosecution History for the 08/689,606 Patent Application filed 8/12/96) (issued as U.S. Patent 5,943,422, incorporating '107), Amendment, 7/2/98, pp. 1-2, 101 (MSI188164-165, MSI188264)</p> <p>4. Rhoads, U.S. Patent No. 5,636,292:</p> <p>a. "Fully Exact Steganography</p> <p>Prior art steganographic methods currently known to the inventor generally involve fully deterministic or 'exact' prescriptions for passing a message. Another way to say this is that it is a basic assumption that for a given message to be passed correctly in its entirety, the receiver of the information needs to receive the exact digital data file sent by the sender, tolerating no bit errors or 'loss' of data. By definition, 'lossy' compression and decompression on empirical signals defeat such steganographic methods. (Prior art, such as the previously noted Komatsu work, are the exceptions here.)</p> <p>The principles of this invention can also be utilized as an exact form of steganography proper. It is suggested that such exact forms of steganography, whether those of prior art or those of this invention, be combined with the relatively recent art of the 'digital signature' and/or the DSS (digital signature standard) in such a way that a receiver of a given empirical data file can first verify that not one single bit of information has been altered in the received file, and thus verify that the contained exact steganographic message has not been altered." (Rhoads 55:5-26)</p> <p>b. "One exemplary application is placement of identification recognition units directly within modestly priced home audio and video instrumentation (such as a TV). Such recognition units would typically monitor 'audio and/or video looking for these copyright identification codes, and thence triggering simple decisions based on the findings, such as disabling or enabling recording capabilities, or incrementing program specific billing meters which are transmitted back to a central audio/video service provider and placed onto monthly invoices." (Rhoads 29:23-33)</p> <p>5. "Use of secure electronic containers to transport items provides an unprecedented degree of security, trustedness and flexibility." ('683 8:50-52)</p> <p>6. "Even if the object is stored locally to the VDE node, it may be stored as a secure or protected object so that it is not directly accessible to a calling process. ACCESS method 2000 establishes the connections, routings, and security</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>requisites needed to access the object.” (‘193 192:14-19)</p> <p>7. “Electronic delivery person 4060 receives item 4054 in digital form and places it into a secure electronic container 302—thus forming a digital ‘object’ 300. A digital object 300 may in this case be, for example, as shown in FIGS. 5A and 5B, and may include one or more containers 302 containing item 4054. FIG. 88 illustrates secure electronic container 302 as an attaché case handcuffed to the secure delivery person’s wrist. Once again, container is shown as a physical thing for purposes of illustration only—in the example it is preferably electronic rather than physical, and comprises digital information having a well-defined structure (see FIG. 5A). Special mathematical techniques known as ‘cryptography’ can be used to make electronic container 302 secure so that only intended recipient 4056 can open the container and access the electronic document (or other item) 4054 it contains.” (‘683 15:56 - 16:6)</p> <p>8. “[C]ontainer 152 can only be opened within a secure protected processing environment 154 that is part of the virtual distribution environment described in the above-referenced Ginter et al. patent disclosure ....” (‘712 168:22-25)</p> <p>9. “A VDE content container is an object that contains both content (for example, commercially distributed electronic information products such as computer software programs, movies, electronic publications or reference materials, etc.) and certain control information related to the use of the object’s content.” (‘193 19:15-21)</p> <p>10. “Other applications, such as application 608b shown in FIG. 11B, may not be ‘VDE Aware’ and therefore may not ‘know’ how to directly access an interface to VDE functions 604 provided by API 682. To provide for this, ROS 602 may include a ‘redirector’ 684 that allows such ‘non- VDE aware’ applications 608(b) to access VDE objects 300 and functions 604. Redirector 684, in the preferred embodiment, translates OS calls directed to the ‘other OS functions’ 606 into calls to the ‘VDE functions’ 604. As one simple example, redirector 684 may intercept a ‘file open’ call from application 608(b), determine whether the file to be opened is contained within a VDE container 300, and if it is, generate appropriate VDE function call(s) to file system 687 to open the VDE container (and potentially generate events to HPE 655 and/or SPE 503 to determine the name(s) of file(s) that may be stored in a VDE object 300, establish a control structure associated with a VDE object 300, perform a registration for a VDE object 300, etc.). Without redirector 684 in this example, a non-VDE aware application such as 608b could access only the part of API 682 that provides an interface to other OS functions 606, and therefore could not access any VDE functions.” (‘193 82:24-45)</p> <p>11. “ACCESS method 2000 reads the ACCESS method MDE from the secure database, reads it in accordance with the ACCESS method DTD, and loads encrypted content source and routing information based on the MDE (blocks 2010, 2012). This source and routing information specifies the location of the encrypted content. ACCESS method 2000 then determines whether a connection to the content is available (decision block 2014). This ‘connection’ could be, for example, an on-line connection to a remote site, a real-time information feed, or a path to a secure/protected resource, for example. If the connection to the content</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>is not currently available ('No' exit of decision block 2014), then ACCESS method 2000 takes steps to open the connection (block 2016). If the connection fails (e.g., because the user is not authorized to access a protected secure resource), then the ACCESS method 2000 returns with a failure indication (termination point 2018)." ('193 192:36-52)</p> <p>12. "Appliance 600B may deliver the digital copy of item 4054 within a container 302 and/or may protect the item with seals, electronic fingerprints, watermarks and/or other visible and/or hidden markings to provide a 'virtual container' or some of the security or other characteristics of a container (for example, the ability to associate electronic controls with the item)." ('683 18:49-56)</p> <p>13. "Trade-offs between flexibility, ease of use and incompatibility and interoperability can be further complicated when security considerations come into play. To be effective in many electronic commerce applications, electronic container designs should be tamper-resistant and secure. One must assume that any tools widely used to create and/or use containers will fall into the hands of those trying to break or crack open the containers or otherwise use digital information without authorization. Therefore, the container creation and usage tools must themselves be secure in the sense that they must protect certain details about the container design. This additional security requirement can make it even more difficult to make containers easy to use and to provide interoperability." ('861 4:51-64)</p> <p><b>Extrinsic:</b></p> <p>1. Container: "VDE objects are represented in a special form called a container. The container is implemented within the VDE as an object-oriented container class. The container class provides a standard method by which applications software may encapsulate and read information stored within the object. Additionally, the container may include procedural information associated with the data being stored. Containers may be nested, and share attributes with nested elements. Nested containers are stored within a larger container. VDE recognizes the presence of additional objects within the content, and allows the nested containers to share, extend or override the attributes of an outer container." (VDE ROI DEVICE v1.0a, 2/9/94, IT00008572)</p> <p>2. Secure: "Pertaining to the control of who can use an object and to the extent to which the object can be used by controlling the authority given to the user." (IBM)</p> <p>3. Container: "In data security, a multilevel information structure. A container has a classification and may contain objects and/or other containers." (Longley)</p> <p>4. Container: "A protected (encrypted) storage object that incorporates descriptive information, protected content, and (optionally) control objects applicable to that content." (IT Glossary, 3/7/95, IT00709617)</p> <p>5. Container: "A contains protected <i>content</i>, which is divided into one or more <i>atomic elements</i>, and, optionally, <i>PERCs</i> governing the <i>content</i> and may be manipulated only as specified by a <i>PERC</i>." (IT Glossary, 4/6/95, IT00028206)</p> <p>6. Container: "A packaging mechanism, consisting of: *One or more Element-derived components. *An organization mechanism which provides a unique name</p> |

|     | Claim<br>Term/Phrase                                  | Evidence Supporting MS Construction  |
|-----|---|--|
|     |   | <p>within a flat namespace for each of the components in a Container.” (IT Glossary, 5/12/95, IT00028293)</p> <p>7. Container: “A protected digital information storage and transport mechanism for packaging content and control information.” (IT Glossary, 8/21/95, TD00068B, IT00032372)</p> <p>8. Secure container: “‘Secure Container(s)’ means electronic container(s) or electronic data arrangements that: (I) use one or more cryptographic or other obfuscation techniques to provide protection for at least a portion of the Content thereof; and (ii) supports the use of Rules and Controls to enable the Management of Content.” (License Agreement IT and Universal Music Group, 4/13/99, Exhibit 11 to IT 30(b)(6))</p> <p>9. Secure container: “A DigiBox container provides security through encryption and the PPE of a commerce node. A secure container does not require a secure communications transport mode.” (IT00035965)</p> <p>10. “A DigiBox container provides for the persistent protection of its properties.” (IT 00035920)</p> <p>11. “DigiBox containers ensure integrity.” (IT00035895)</p>  |
| 21. | <p>tamper<br/>resistance</p> <p>721.1</p>             | <p><b>Intrinsic:</b></p> <p>1. “The level of security and tamper resistance required for trusted SPU hardware processes depends on the commercial requirements of particular markets or market niches, and may vary widely.” (‘193 49:59-62)</p> <p><b>Extrinsic:</b></p> <p>1. Tamper-resistant Module: “In data security, a device in which sensitive information, such as a master cryptographic key, is stored and cryptographic functions are performed. The device has one or more sensors to detect physical attacks, by an adversary trying to gain access to the stored information in which case the stored sensitive data is immediately destroyed.” (Longley)</p> <p>2. See also IT41530-49, IT51147-60</p> <p>3. “Subversion: A compromise that undermines integrity.” (Neumann, p. 349)</p> <p>4. “Spoofing: Taking on the characteristics of another system or used for purposes of deception. In the present contexts, spoofing is generally prankish rather than overtly malicious, although it is often used elsewhere in a malicious contexts.” (Neumann, p. 349)</p> <p>5. Security: “1. Protection against unwanted behaviors. In present usage, computer security includes properties such as confidentiality, integrity, availability, prevention of denial of service, and prevention of generalized misuse. 2. The property that a particular security policy is enforced, with some degree of assurance. 3. Security is sometimes used in the restricted sense of confidentiality, particularly in the case of multilevel security (that is, multilevel confidentiality).” (Neumann, p. 349)</p> |
| 22. | <p>tamper<br/>resistant<br/>barrier</p> <p>721.34</p> | <p><b>Intrinsic:</b></p> <p>1. “In addition, Applicants would like to draw the Examiner’s attention to other sections of the specification in support of words or phrases cited by the Examiner as ‘indefinite.’ ... In claims ... 36 ... the term ‘barrier’ is used as part of the phrase ‘tamper resistant barrier.’ This phrase is described in the specification on</p>  |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>at least pages 7-8 and 46. In addition, the incorporated Ginter application describes tamper resistant barriers in a number of locations such as, for example, page 201." (Prosecution History for the 08/689,754 Patent Application (issued as the '721), Amendment, 4/14/99, p. 14.) (p. 7 and 46 of the original specification are '721 2:62 - 3:13 and 16:35-54 of the issued patent; p. 201 of Ginter application 08/388,107 is '193 80:40 - 81:1)</p> <ol style="list-style-type: none"> <li>2. "SPU 500 is enclosed within and protected by a 'tamper resistant security barrier' 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions." ('193 59:48-53)</li> <li>3. "Although block 1262 includes encrypted summary services information on the back up, it preferably does not include SPU device private keys, shared keys, SPU code and other internal security information to prevent this information from ever becoming available to users even in encrypted form." ('193 166:59-64)</li> <li>4. "Briefly, the preferred example software-based PPE 650 installation process provides the following security techniques: encrypted software distribution, installation customized on a unique instance and/or electronic appliance basis, encrypted on-disk form, installation tied to payment method, unique software and data layout, and identifiable copies." ('900 236:32-42)</li> <li>5. "... (c) if the load module has an associate digital signature, authenticating the digital signature at least one public key secured behind a tamper resistant barrier and therefore hidden from the user." ('721 22:5-16 (claim 9))</li> <li>6. "A further attack technique might involve duplicating one installed operational material 3472 instance by copying the programs and data from one personal computer 3372B to another personal computer 3372C or emulator (see FIG. 67B, block 3364, and the 'copy' arrow 3364A in FIG. 67A). The duplicated PPE instance could be used in a variety of ways, such as, for example, to place an imposter PPE 650 instance on-line and/or to permit further dynamic analysis." ('900 233:8-15)</li> <li>7. "Various software protection techniques detailed above in connection with FIG. 10 may provide software-based tamper resistant barrier 674 within a software-only and/or hybrid software/hardware protected processing environment 650. The following is an elaboration on those above-described techniques. These software protection techniques may provide, for example, the following: An on-line registration process that results in the creation of a shared secret between the registry and the PPE 650 instance—used by the registry to create content and transactions that are meaningful only to specific PPE instance. An installation program (that may be distinct from the PPE operational material software) that creates a customized installation of the PPE software unique to each PPE instance and/or associate electronic appliance 600. Camouflage protections that make it difficult to reverse engineer the PPE 650 operational materials during PPE 650 operation. Integrity checks performed during PPE 650 operation (e.g., during on-line interactions with trusted servers) to detect compromise. In general, the software-based tamper resistant barrier 674 may establish 'trust' primarily through uniqueness and complexity." ('900 235:30-57)</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>8. "Operational materials 3472 may then decrypt the next program segment dynamically ... This mechanism increases the tamper-resistant of the executable code-- thus providing additional tamper resistance for PPE operations." ('900 243:3-9)</p> <p>9. "The software-based tamper resistant barrier 674 provided by HPE 655 may be provided, for example, by: introducing time checks and/or code modifications to complicate the process of stepping through code comprising a portion of kernel 688a and/or a portion of component assemblies 690 using a debugger; using a map of defects on a storage device (e.g., a hard disk, memory card, etc.) to form internal test values to impede moving and/or copying HPE 655 to other electronic appliances 600; using kernel code that contains false branches and other complications in flow of control to disguise internal processes to some degree from disassembly or other efforts to discover details of processes; using 'self-generating' code (based on the output of a co-sine transform, for example) such that detailed and/or complete instruction sequences are not stored explicitly on storage devices and/or in active memory but rather are generated as needed; using code that 'shuffles' memory locations used for data values based on operational parameters to complicate efforts to manipulate such values; using any software and/or hardware memory management resources of electronic appliance 600 to 'protect' the operation of HPE 655 from other processes, functions, etc. Although such a software-based tamper resistant barrier 674 may provide a fair degree of security, it typically will not be as secure as the hardware-based tamper resistant barrier 502 provided (at least in part) by SPU 500." ('193 80:40-65, Fig. 10)</p> <p>10. "Software-based tamper resistant barrier 674 may be created by software executing on a general-purpose CPU. Various software protection techniques may be used to construct and/or provide software-based tamper resistant barrier 674." ('900 230:61-65)</p> <p>11. "No software-only tamper resistant barrier 674 can be wholly effective against all of these threats. A sufficiently powerful dynamic analysis (such as one employing an in-circuit emulator) can lay bare all of the software-based PPE 650's secrets. Nonetheless, various techniques described below in connection with FIG. 69A and following make such an analysis extremely frustrating and time consuming--increasing the 'work factor' to a point where it may become commercially unfeasible to attempt to 'crack' a software-based tamper resistant barrier 674." ('900 233:24-33)</p> <p>12. "For example, the PPE 650 may rewrite or overwrite memory locations immediately after using same to make their contents unavailable for scrutiny. Similarly, the PPE 650 operational software may use hardware and/or time dependent sequences to prevent emulation. Additionally, some of the PPE 650 environment code may be self-modifying." ('900 236:9-15)</p> <p><b>Extrinsic:</b></p> <p>1. Tamper-resistant module: "In data security, a device in which sensitive information, such as a master cryptographic key, is stored and cryptographic functions are performed. The device has one or more sensors to detect physical attacks, by an adversary trying to gain access to the stored information in which</p> |



|     | Claim Term/Phrase  | Evidence Supporting MS Construction  |
|-----|--|--|
|     |  | <p>case the stored sensitive data is immediately destroyed.” (Longley)</p> <p>2. “The ‘tamper-resistant module’ is physically strong and destroys secrets when opened, and the software running inside has been checked for integrity;” (Davies, p. 3)</p> <p>3. “The host computer is provided with a specially, physically secure module containing all the secret information which must be protected. In the IBM papers it is called the ‘Cryptographic Facility’: we shall call it a ‘Tamper Resistant Module’ (TRM).” (Davies, p. 144)</p>   |
| 23. | <p>use</p> <p>193.19</p> <p>683.2</p> <p>721.1</p> <p>861.58</p> <p>891.1</p> <p>912.8, 912.35</p> | <p><b>Intrinsic:</b></p> <p>1. “Provides non-repudiation of use and may record specific forms of use such as viewing, editing, extracting, copying, redistributing (including to what one or more parties), and/or saving.” (‘683 6:46-48)</p> <p>2. “Content (executables for example) delivered with proof of delivery and/or execution or other use.” (‘683 7:8-9)</p> <p>3. “In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.” (‘193 6:24-31)</p> <p>4. “Some or all of the back up files may be packaged within an administrative object and transmitted for analysis, transportation, or other uses.” (‘193 167:45-48)</p> <p>5. “to securely control access and other use, including distribution of records, documents, and notes associated with the case.” (‘193 274:34-36)</p> <p>6. “Thus wrapped, a VDE object may be distributed to the recipient without fear of unauthorized access and/or other use. The one or more authorized users who have received an object are the only parties who may open that object and view and/or manipulate and/or otherwise modify its contents and VDE secure auditing ensures a record of all such user content activities.” (‘193 277:15-21)</p> <p>7. “These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc.” (‘193 9:24-27)</p> <p>8. “VDE provides a secure, distributed electronic transaction management system for controlling the distribution and/or other usage of electronically provided and/or stored information.” (‘193 9:36-39)</p> <p>9. “As a result, VDE supports most types of electronic information and/or appliance: usage control (including distribution), security, usage auditing, reporting, other administration, and payment arrangements.” (‘193 13:50-53)</p> <p>10. “SPU 500 is enclosed within and protected by a ‘tamper resistant security barrier’ 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500. In one example, tamper resistant security barrier 502 is formed by security features such as ‘encryption,’ and hardware that detects tampering and/or destroys sensitive information within secure environment 503 when tampering is detected.” (‘193 59:48-59)</p> |

|     | Claim<br>Term/Phrase   | Evidence Supporting MS Construction  |
|-----|--|--|
|     |  | <p>11. "Once the information is downloaded, the now-initialized PPE 650 can discard (or simply not use) the manufacturing key." ('193 212:57-59)</p> <p><b>Extrinsic:</b></p> <ol style="list-style-type: none"> <li>1. User: "A person using a InterTrust node to perform some function (i.e., acting in some role). A user is identified with respect to the node by a user ID." (IT Glossary, 5/12/95, IT00028300)</li> <li>2. User ID: "Locally to a InterTrust node, each InterTrust user has an ID associated with a user name and authentication (e.g., password). In some deployments, there may be only one user, and access to the machine may be considered sufficient authentication; in such cases, the user ID concept may not be visible to the user even though it is present in the implementation." (IT Glossary, 5/12/95, IT00028301)</li> <li>3. Use: "To use an object is to access the content. This involves the processes of controlling and metering the use of the property and creating audit trail records on the use." (VDE ROI DEVICE v1.0a, 2/9/94, IT00008570)</li> </ol>  |
| 24. | <p>virtual distribution environment</p> <p>900.155</p> <p>Also as set forth in each "claim as a whole" by Microsoft.</p> | <p><u>Virtual Distribution Environment:</u></p> <p><u>"CLAIM AS A WHOLE":</u></p> <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. "The instant application is one of a series of applications which are all generally directed to a virtual distribution environment." (09/208,017 ('193), Examiner's Amendment, 8/4/00, p. 2)</li> <li>2. See generally Background and Summary of Invention of '193 Patent ('193 2:22 - 49:63)</li> <li>3. "With respect to the remaining issues, Applicants respectfully disagree. For example, the Examiner objects to the use of 'environment' as indefinite and unclear. This word, however, is not used in isolation, but rather in the context of several longer phrases, all of which are defined in the specification. The phrase 'protected processing environment,' for example, is used in Claims 11 and 15-18 and described on at least, for example, pages 7-8 and 25 of the specification. The term 'virtual distribution environment' used in Claim 11 is described, for example, on page 7 of the specification. The terms are also described in the commonly copending application Serial Number 08/388,107 of Ginter et al., filed 13 February 1995, entitled 'System and Methods for Secure Transaction Management and Electronic Rights Protection.' A copy of the incorporated Ginter application can be provided to the Examiner upon request." 08/689,754 ('721), Amendment, 4/14/99, p. 13 (pp. 7, 7-8 and 25 of the original specification are '721 2:62 - 3:13, 2:62 - 3:34 and 8:6-28 of the issued patent)</li> <li>4. See also, Prosecution History of '900:</li> </ol> <p>Claims 302, 321 and 322, as pending:</p> <p>"302. A virtual distribution environment comprising</p> <ul style="list-style-type: none"> <li>• a first host processing environment comprising</li> <li>• a central processing unit;</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <ul style="list-style-type: none"> <li>• main memory operatively connected to said central processing unit;</li> <li>• mass storage operatively connected to said central processing unit and said main memory;</li> <li>• said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising: <ul style="list-style-type: none"> <li>• machine check programming which derives information from one or more aspects of said host processing environment,</li> <li>• one or more storage locations storing said information; and</li> <li>• integrity programming which <ul style="list-style-type: none"> <li>• causes said machine check programming to derive said information,</li> <li>• compares said information to information previously stored in said one or more storage locations, and</li> <li>• generates an indication based on the result of said comparison.</li> </ul> </li> </ul> </li> </ul> <p>321. A virtual distribution environment as in claim 302,</p> <ul style="list-style-type: none"> <li>• said virtual distribution environment further comprising programming which takes one or more actions based on the state of said indication.</li> </ul> <p>322. A virtual distribution environment as in claim 321 in which said one or more actions includes at least temporarily halting further processing.”<br/>(Prosecution History for Patent Application 08/706,206 (issued as the ‘900 patent), Amendment, 06/09/98, 92-93, 96, 96-97))</p> <p>b. “Claims ... 322-324, ... are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.”<br/>(Prosecution History for Patent Application 08/706,206, Office Action, 08/27/98, p. 2)</p> <p>c. “322. A virtual distribution environment comprising</p> <ul style="list-style-type: none"> <li>• a first host processing environment comprising <ul style="list-style-type: none"> <li>• a central processing unit;</li> <li>• main memory operatively connected to said central processing unit;</li> <li>• mass storage operatively connected to said central processing unit and said main memory;</li> <li>• said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising: <ul style="list-style-type: none"> <li>• machine check programming which derives information from one or more aspects of said host processing environment,</li> <li>• one or more storage locations storing said information;</li> <li>• integrity programming which <ul style="list-style-type: none"> <li>o causes said machine check programming to derive said information,</li> <li>o compares said information to information previously stored in said one or more storage locations, and</li> <li>o generates an indication based on the result of said comparison; and</li> </ul> </li> </ul> </li> <li>• programming which takes one or more actions based on the state of</li> </ul> </li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>said indication;</p> <ul style="list-style-type: none"> <li>• said one or more actions including at least temporarily halting further processing.” ... Remarks, “Applicants appreciate the indication that claims ... are allowed and that claims ... 322-324 are objected to but would be allowable if rewritten into independent form. ... For purposes of expedition, applicants are cancelling the rejected claims without prejudice ..., and are rewriting objected to dependent claims into independent form.” (Prosecution History for Patent Application 08/706,206, Amendment, 11/23/98, p. 27-28, 42)</li> </ul> <p>(1) <u>DATA SECURITY AND COMMERCE WORLD:</u></p> <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “VDE supports a model wide, distributed security implementation which creates a single secure ‘virtual’ transaction processing and information storage environment. VDE enables distributed VDE installations to securely store and communicate information and remotely control the execution processes and the character of use of electronic information at other VDE installations and in a wide variety of ways. . . .” (‘193 21:57-65)</li> <li>2. “The rights protection problems solved by the present invention are electronic versions of basic societal issues. These issues include protecting property rights, protecting privacy rights, properly compensating people and organizations for their work and risk, protecting money and credit, and generally protecting the security of information.” (‘193 4:8-13)</li> <li>3. “The present invention provides a new kind of ‘virtual distribution environment’ (called ‘VDE’ in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels ‘across’ the ‘information highway.’” (‘193 2:24-28)</li> <li>4. “A fundamental problem for electronic content providers is extending their ability to control the use of proprietary information. Content providers often need to limit use to authorized activities and amounts. Participants in a business model involving, for example, provision of movies and advertising on optical discs may include actors, directors, script and other writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, and content end-users. These participants need the ability to embody their range of agreements and requirements, including use limitations, into an ‘extended’ agreement comprising an overall electronic business model. This extended agreement is represented by electronic content control information that can automatically enforce agreed upon rights and obligations. Under VDE, such an extended agreement may comprise an electronic contract involving all business model participants. Such an agreement may alternatively, or in addition, be made up of electronic agreements between subsets of the business model participants. Through the use of VDE, electronic commerce can function in the same way as traditional commerce-that</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>is commercial relationships regarding products and services can be shaped through the negotiation of one or more agreements between a variety of parties.” (‘193 2:37-60)</p> <p>5. “Protecting the rights of electronic community members involves a broad range of technologies. VDE combines these technologies in a way that creates a ‘distributed’ electronic rights protection ‘environment.’ This environment secures and protects transactions and other processes important for rights protection. VDE, for example, provides the ability to prevent, or impede, interference with and/or observation of, important rights related transactions and processes.” (‘193 3:63 - 4:3)</p> <p>6. “VDE is a cost-effective and efficient rights protection solution that provides a unified, consistent system for securing and managing transaction processing. VDE can: (a) audit and analyze the use of content, (b) ensure that content is used only in authorized ways, and (c) allow information regarding content usage to be used only in ways approved by content users.” (‘193 4:48-55)</p> <p>7. “In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.” (‘193 6:24-30)</p> <p>8. “A variety of capabilities are required to implement an electronic commerce environment. VDE is the first system that provides many of these capabilities and therefore solves fundamental problems related to electronic dissemination of information.” (‘193 8:16-20)</p> <p>9. “VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment. VDE is not limited to being an application or application specific toolset that covers only a limited subset of electronic interaction activities and participants. Rather, VDE supports systems by which such applications can be created, modified, and/or reused. As a result, the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components. VDE can support a single electronic ‘world’ within which most forms of electronic transaction activities can be managed.” (‘193 8:53 - 9:5)</p> <p>10. “VDE can securely manage the integration of control information provided by two or more parties. As a result, VDE can construct an electronic agreement between VDE participants that represent a ‘negotiation’ between, the control requirements of, two or more parties and enacts terms and conditions of a resulting agreement. VDE ensures the rights of each party to an electronic agreement regarding a wide range of electronic activities related to electronic</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>information and/or appliance usage.” (‘193 9:52-61)</p> <ol style="list-style-type: none"> <li>11. “‘Hardware’ 506 also contains long-term and short-term memories to store information securely so it can’t be tampered with.” (‘193 60:1-3)</li> <li>12. “VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information.” (‘193 11:60-63)</li> <li>13. “Together, these VDE components comprise a secure, virtual, distributed content and/or appliance control, auditing (and other administration), reporting, and payment environment.” (‘193 13:14-17)</li> <li>14. “VDE can securely deliver information from one party to another concerning the use of commercially distributed electronic content. Even if parties are separated by several ‘steps’ in a chain (pathway) of handling for such content usage information, such information is protected by VDE through encryption and/or other secure processing. Because of that protection, the accuracy of such information is guaranteed by VDE, and the information can be trusted by all parties to whom it is delivered.” (‘193 14:31-39)</li> <li>15. “VDE allows the needs of electronic commerce participants to be served and it can bind such participants together in a universe wide, trusted commercial network that can be secure enough to support very large amounts of commerce. VDE’s security and metering secure subsystem core will be present at all physical locations where VDE related content is (a) assigned usage related control information (rules and mediating data), and/or (b) used. This core can perform security and auditing functions (including metering) that operate within a ‘virtual black box,’ a collection of distributed, very secure VDE related hardware instances that are interconnected by secured information exchange (for example, telecommunication) processes and distributed database means.” (‘193 15:14-27)</li> <li>16. “VDE provides organization, community, and/or universe wide secure environments whose integrity is assured by processes securely controlled in VDE participant user installations (nodes).” (‘193 20:48-51)</li> <li>17. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention: VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, ... employ ‘templates’ to ease the process of configuring capabilities of the present invention as they relate to specific industries or businesses.... Given the very large range of capabilities and configurations supported by the present invention, reducing the range of configuration opportunities to a manageable subset particularly appropriate for a given business model allows the full configurable power of the present invention to be easily employed by ‘typical’ users who would be otherwise burdened with complex programming and/or configuration design responsibilities template applications can also help ensure that VDE related processes are secure and optimally bug free by reducing the risks associated with the contribution of independently developed load modules, including unpredictable aspects of code interaction between independent modules and applications, as well as security</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>risks associated with possible presence of viruses in such modules.... As the context surrounding these templates changes or evolves, template applications provided under the present invention may be modified to meet these changes for broad use, or for more focused activities .... Of course, templates may, under certain circumstances have fixed control information and not provide for user selections or parameter data entry.” (‘193 21:43-53; 27:1 - 28:18)</p> <p>18. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention:... provide mechanisms to persistently maintain trusted content usage and reporting control information through both a sufficiently secure chain of handling of content and content control information and through various forms of usage of such content wherein said persistence of control may survive such use. Persistence of control includes the ability to extract information from a VDE container object by creating a new container whose contents are at least in part secured and that contains both the extracted content and at least a portion of the control information which control information of the original container and/or are at least in part produced by control information of the original container for this purpose and/or VDE installation control information stipulates should persist and/or control usage of content in the newly formed container. Such control information can continue to manage usage of container content if the container is ‘embedded’ into another VDE managed object, such as an object which contains plural embedded VDE containers, each of which contains content derived (extracted) from a different source.” (‘193 21:43-45; 28:45-65)</p> <p>19. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... Interoperability is fundamental to efficient electronic commerce. The design of the VDE foundation, VDE load modules, and VDE containers, are important features that enable the VDE node operating environment to be compatible with a very broad range of electronic appliances.” (‘193 21:43-45; 34:25-30)</p> <p>20. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... securely support electronic currency and credit usage control, storage, and communication at, and between, VDE installations.” (‘193 21:43-45; 36:49-51)</p> <p>21. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... requiring reporting and payment compliance by employing exhaustion of budgets and time ageing of keys.” (‘193 21:43-45; 40:8-9)</p> <p>22. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... Because of the VDE security, including use of effective encryption, authentication, digital signaturing, and secure database structures, the records contained within a VDE card arrangement may be accepted as valid transaction records for government and/or corporate recordkeeping requirements.” (‘193 21:43-45; 41:37-42)</p> <p>23. “Since all secure communications are at least in part encrypted and the processing inside the secure subsystem is concealed from outside observation and interference, the present invention ensures that content control information can be enforced.” (‘193 46:4-8)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>24. "An important feature of VDE is that it can be used to assure the administration of, and adequacy of security and rights protection for, electronic agreements implemented through the use of the present invention." ('193 46:51-54)</p> <p>25. "These are merely a few simple examples demonstrating the importance of ROS 602 ensuring that certain component assemblies 690 are formed in a secure manner. ROS 602 provides a wide range of protections against a wide range of 'threats' to the secure handling and execution of component assemblies 690." ('193 85:15-20)</p> <p>26. "VDE further enables this process by providing a secure execution space in which the negotiation process(es) are assured of integrity and confidentiality in their operation." ('193 245:20-22)</p> <p>27. "Taken together, and employed at times with VDE administrative objects and VDE security arrangements and processes, the present invention truly achieves a content control and auditing architecture that can be configured to most any commercial distribution embodiment." ('193 261:10-15)</p> <p>28. "For example, VDE 100 positively controls content access and usage, provides guarantee of payment for content used, and enforces budget limits for accessed content." ('193 240:53-56)</p> <p>29. "Such metering is a flexible basis for ensuring payment for content royalties, licensing, purchasing, and/or advertising." ('193 33:56-58)</p> <p>30. "The overall integrity and security of VDE 100 could ensure, in a coherent and centralized manner, that electronic reporting of tax related information (derived from one or more electronic commerce activities) would be valid and comprehensive." ('193 237:47-51)</p> <p>31. "Distributors 106 and financial clearinghouses 116 may themselves be audited based on secure records of their administrative activities and a chain of reliable, 'trusted' processes ensures the integrity of the overall digital distribution process. This allows content owners, for example, to verify that they are receiving appropriate compensation based on actual content usage or other agreed-upon bases." ('193 254:66 - 255:5)</p> <p>32. "Because the control information is carried with each copy of a VDE protected document, and can ensure that central registries are updated and/or that originators are notified of document use, tracking can be prompt and accurate." ('193 281:14-16)</p> <p>33. "A final desirable feature of agreements in general (and electronic representations of agreements in particular) is that they be accurately recorded in a non-repudiatable form. In traditional terms, this involves creating a paper document (a contract) that describes the rights, restrictions, and obligations of all parties involved. This document is read and then signed by all parties as being an accurate representation of the agreement. Electronic agreements, by their nature, may not be initially rendered in paper. VDE enables such agreements to be accurately electronically described and then electronically signed to prevent repudiation." ('193 245:25-35)</p> <p>34. "As discussed above, a wide variety of techniques are currently being used to provide secure, trusted confidential delivery of documents and other items. Unfortunately, none of these previously existing mechanisms provide truly</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>trusted, virtually instantaneous delivery on a cost-effective, convenient basis and none provide rights management and auditing through persistent, secure, digital information protection.</p> <p>In contrast, the present inventions provide the trustedness, confidentiality and security of a personal trusted courier on a virtually instantaneous and highly cost-effective basis. They provide techniques, systems and methods that can be applied to any form of electronic communications (including, but not limited to Internet and internal company electronic mail) an extremely high degree of trustedness, confidence and security approaching or exceeding that provided by a trusted personal courier. They also provide a wide variety of benefits that flow from rights management and secure chain of handling and control." ('683 5:22-40)</p> <p>35. "The Virtual Distribution Environment provides comprehensive overall systems, and wide arrays of methods, techniques, structures and arrangements, that enable secure, efficient electronic commerce and rights management on the Internet and other information superhighways and on internal corporate networks such as 'Intranets'." ('683 5:41-51-56)</p> <p>36. "Parties using the Virtual Distribution Environment can participate in commerce and other transactions in accordance with a persistent set of rules they electronically define." ('683 6:11-14)</p> <p>37. "All of these various coordination steps can be performed nearly simultaneously, efficiently, rapidly and with an extremely high degree of trustedness based on the user of electronic containers 302 and the secure communications, authentication, notarization and archiving techniques provided in accordance with the present inventions." ('683 55:54-59)</p> <p>38. "People are increasingly using secure digital containers to safely and securely store and transport digital content. One secure digital container model is the 'DigiBox™' container developed by InterTrust Technologies, Inc. of Sunnyvale, Calif. The Ginter et al. patent specification referenced above describes many characteristics of this DigiBox™ container model—a powerful, flexible, general construct that enables protected, efficient and interoperable electronic description and regulation of electronic commerce relationship of all kinds, including the secure transport, storage and rights management interface with objects and digital information within such containers." ('861 1:35-41)</p> <p>39. "Briefly, DigiBox containers are tamper-resistant digital containers that can be used to package any kind of digital information such as, for example, text, graphics, executable software, audio and/or video. The rights management environment in which DigiBox™ containers are used allows commerce participants to associate rules with the digital information (content). The rights management environment also allows rules (herein including rules and parameter data controls) to be securely associated with other rights management information, such as for example, rules, audit records created during use of digital information and administrative information associated with keeping the environment working properly, including ensuring rights and any agreements among parties. The DigiBox™ electronic container can be used to store, transport and provide a rights management interfaces to digital information, related rules and other rights management information, as well as to other objects</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>and/or data within a distributed, rights management environment. This arrangement can be used to provide electronically enforced chain of handling and control wherein rights management persists as a container moves from one entity to another. This capability helps support a digital rights management architecture that allows content rightsholders (including any parties who have system authorized interests related to such content, such as content republishes or even governmental authorities) to securely control and manage content, events, transactions, rules and usage consequences, including any required payment and/or usage reporting. This secure control and management continues persistently, protecting rights as content is delivered to, used by, and passed among creators, distributors, repurposes, consumers, payment disaggregators, and other value chain participants.” (‘861 1:47 - 2:12)</p> <p>40. “Use of secure electronic containers to transport items provides an unprecedented degree of security, trustedness and flexibility.” (‘683 8:50-52)</p> <p>41. “Virtual distribution environment 100 is ‘virtual’ because it does not require many of the physical ‘things’ that used to be necessary to protect rights, ensure reliable and predictable distribution, and ensure proper compensation to content creators and distributors.” (‘193 53:23-27)</p> <p><b>Extrinsic:</b></p> <p>42. VDE: “VDE is the broad name given to a comprehensive system (algorithms, software, and hardware) that provides metering, securing, and administration tools for intellectual property. VDE stands for ‘Virtual Distribution Environment.’” (VDE ROI DEVICE v1.0a, 2/9/94, IT00008570)</p> <p>43. Virtual: “Pertaining to a functional unit that appears to be real, but whose functions are accomplished by other means.” (IBM)</p> <p>44. Environment: “1. The aggregate of external circumstances, conditions, and objects that affect the development, operation, and maintenance of a system. 2. In computer security, those factors, both internal and external, of an ADP system that help to define the risks associated with its operation.” (Longley)</p> <p>45. Environment: See InterTrust node: “A computer that is enabled for processing of DigiBox containers by installation of a PPE, which may be either hardware or software based. A node may include application software and/or operating system integration. The node is also termed the <i>environment</i>.” (IT Glossary, 8/21/95, TD00068B, IT00032375)</p> <p>46. InterTrust Commerce Architecture model: “A model that defines a general-purpose distributed architecture for secure electronic commerce and digital rights management. The InterTrust Commerce Architecture model includes four key software elements: DigiBox secure containers, InterRights Point software with associated protected database, the InterTrust Transaction Authority Framework, and the InterTrust Deployment Manager.” (IT Glossary, 1997, ML00012A)</p> <p>47. VDE is a system using secure computing technology to enforce a chain of handling and control representing the rights of interested parties. (IT Glossary, 3/7/95, IT00709616)</p> <p>48. Virtual Distribution Environment (VDE): “A set of components that protects content and enforces rights associated with content.” (IT Glossary, 3/7/95,</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>IT00709620)</p> <p>49. "Virtual Distribution Environment: or 'VDE' shall mean a system which guarantees: (i) that the content creators, publishers, and/or distributors of information receive agreed upon fees for the use of, and/or records of the use of, electronic content; and/or (ii) that stored and/or distributed information will be used only in authorized ways. More particularly, VDE relates to systems for applying controls to, and controlling and/or auditing use of, electronically stored and/or disseminated information." (License Agreement, National Semiconductor and EPR, 3/18/94, Exhibit 12 to IT 30(b)(6))</p> <p>50. See also IT0001689-96, IT0709785 (VDE on a Page), IT000202-29</p> <p><u>(2) SECURE PROCESSING ENVIRONMENT:</u></p> <ol style="list-style-type: none"> <li>1. "VDE allows the needs of electronic commerce participants, to be served and it can bind such participants together in a universe wide, trusted commercial network that can be secure enough to support very large amounts of commerce. VDE's security and metering secure subsystem core will be present all physical locations where VDE related contents is (a) assigned usage related control information (rules and mediating data), and/or (b) used. This core can perform security and auditing functions (including metering) that operate within a 'virtual black box,' a collection of distributed, very secure VDE related hardware instances that are interconnected by secured information exchange (for example, telecommunication) processes and distributed database means." ('193 15:14-27)</li> <li>2. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... VDE employs special purpose hardware distributed throughout some or all locations of a VDE implementation: a) said hardware controlling important elements of: content preparation (such as causing such content to be placed in a VDE content container and associating content control information with said content), content and/or electronic appliance usage auditing, content usage analysis, as well as content usage control; and b) said hardware having been designed to securely handle processing load module control activities, wherein said control processing activities may involve a sequence of required control factors" ('193 21:43-45; 22:20-31)</li> <li>3. "Physical facility and user identity authentication security procedures may be used instead of hardware SPUs at certain nodes, such as at an established financial clearinghouse, where such procedures may provide sufficient security for trusted interoperability with a VDE arrangement employing hardware SPUs at user nodes." ('193 45:60-65)</li> <li>4. "An important part of VDE provided by the present invention is the core secure transaction control arrangement, herein called an SPU (or SPUs), that typically must be present in each user's computer, other electronic appliance, or network. SPUs provide a trusted environment for generating decryption keys, encrypting and decrypting information, managing the secure communication of keys and other information between electronic appliances (i.e. between VDE installations and/or between plural VDE instances within a single VDE installation), securely accumulating and managing audit trail, reporting, and budget</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>information in secure and/or non-secure non-volatile memory, maintaining a secure database of control information management instructions, and providing a secure environment for performing certain other control and administrative functions.” (‘193 48:66 - 49:17)</p> <p>5. “A hardware SPU (rather than a software emulation) within a VDE node is necessary if a highly trusted environment for performing certain VDE activities is required.” (‘193 49:15-17)</p> <p>6. “‘Hardware’ 506 also contains long-term and short-term memories to store information securely so it can’t be tampered with.” (‘193 60:1-3)</p> <p>7. “A VDE node’s hardware SPU is a core component of a VDE secure subsystem and may employ some or all of an electronic appliance’s primary control logic, such as a microcontroller, microcomputer or other CPU arrangement. This primary control logic may be otherwise employed for non VDE purposes such as the control of some or all of an electronic appliance’s non-VDE functions. When operating in a hardware SPU mode, said primary control logic must be sufficiently secure so as to protect and conceal important VDE processes. For example, a hardware SPU may employ a host electronic appliance microcomputer operating in protected mode while performing VDE related activities, thus allowing portions of VDE processes to execute with a certain degree of security.” (‘193 49:33-46)</p> <p>8. “As shown FIG. 6 [sic], in the preferred embodiment, an SPU 500 may be implemented as a single integrated circuit ‘chip’ 505 to provide a secure processing environment in which confidential and/or commercially valuable information can be safely processed, encrypted and/or decrypted.” (‘193 63:48-52)</p> <p>9. “SPU 500 is enclosed within and protected by a ‘tamper resistant security barrier’ 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500. In one example, tamper resistant security barrier 502 is formed by security features such as ‘encryption,’ and hardware that detects tampering and/or destroys sensitive information within secure environment 503 when tampering is detected.” (‘193 59:48-59)</p> <p>10. “SPU 500 may be surrounded by a tamper-resistant hardware security barrier 502. Part of this security barrier 502 is formed by a plastic or other package in which an SPU ‘die’ is encased. Because the processing occurring within, and information stored by, SPU 500 are not easily accessible to the outside world, they are relatively secure from unauthorized access and tampering. All signals cross barrier 502 through a secure, controlled path provided by BIU 530 that restricts the outside world’s access to the internal components within SPU 500. The secure, controlled path resists attempts from the outside world to access secret information and resources within SPU 500.” (‘193 63:60 - 64:5)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>(3) <u>VDE CONTROLS</u>: See support as listed for Control (n.) , item #8, above.</p> <ol style="list-style-type: none"> <li>1. "Limited only by the VDE control information employed by content creators, other providers, and other pathway of handling and control participants, VDE allows a 'natural' and unhindered flow of, and creation of, electronic content product models." ('193 297:25-29)</li> <li>2. "Regulation is ensured by control information put in place by one or more parties." ('193 6:34-35)</li> <li>3. "As a result, the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components." ('193 8:62 - 9:3)</li> <li>4. "Independently, securely deliverable, component based control information allows efficient interaction among control information sets supplied by different parties." ('193 10:46-50)</li> <li>5. "A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information (normally in the form of VDE objects containing one or more methods, data, or load module VDE components). This independently delivered control information can be integrated with senior and other pre-existing content control information to securely form derived control information using the negotiation mechanisms of the present invention. All requirements specified by this derived control information must be satisfied before VDE controlled content can be accessed or otherwise used. This means that, for example, all load modules and any mediating data which are listed by the derived control information as required must be available and securely perform their required function." ('193 10:66 - 11:14)</li> <li>6. "Content control information governs content usage according to criteria set by holders of rights to an object's contents and/or according to parties who otherwise have rights associated with distributing such content (such as governments, financial credit providers, and users)." ('193 15:46-48)</li> <li>7. "In part, security is enhanced by object methods employed by the present invention because the encryption schemes used to protect an object can efficiently be further used to protect the associated content control information (software control information and relevant data) from modification." ('193 15:51-55)</li> <li>8. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... Content users, such as end-user customers using commercially distributed content (games, information resources, software programs, etc.), can define, if allowed by senior control information, budgets, and/or other control information, to manage their own internal use of content." ('193 21:43-45; 29:3-8)</li> <li>9. "Summary of Some Important Features Provided by VDE in Accordance With</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>the Present Invention.... support the separation of fundamental transaction control processes through the use of event (triggered) based method control mechanisms. These event methods trigger one or more other VDE methods (which are available to a secure VDE sub-system) and are used to carry out VDE managed transaction related processing. These triggered methods include independently (separably) and securely processable component billing management methods, budgeting management methods, metering management methods, and related auditing management processes. As a result of this feature of the present invention, independent triggering of metering, auditing, billing, and budgeting methods, the present invention is able to efficiently, concurrently support multiple financial currencies (e.g. dollars, marks, yen) and content related budgets, and/or billing increments as well as very flexible content distribution models.” (‘193 21:43-45; 42:21-38)</p> <p>10. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention....support, complete, modular separation of the control structures related to (1) content event triggering, (2) auditing, (3) budgeting (including specifying no right of use or unlimited right of use), (4) billing, and (5) user identity (VDE installation, client name, department, network, and/or user, etc.). The independence of these VDE control structures provides a flexible system which allows plural relationships between two or more of these structures, for example, the ability to associate a financial budget with different event trigger structures (that are put in place to enable controlling content based on its logical portions). Without such separation between these basic VDE capabilities, it would be more difficult to efficiently maintain separate metering, budgeting, identification, and/or billing activities which involve the same, differing (including overlapping), or entirely different, portions of content for metering, billing, budgeting, and user identification, for example, paying fees associated with usage of content, performing home banking, managing advertising services, etc. VDE modular separation of these basic capabilities supports the programming of plural, ‘arbitrary’ relationships between one or differing content portions (and/or portion units) and budgeting, auditing, and/or billing control information.” (‘193 21:43-45; 42:39-63)</p> <p>11. “The virtual distribution environment 100 prevents use of protected information except as permitted by the ‘rules and controls’ (control information). For example, the ‘rules and controls’ shown in FIG. 2 may grant specific individuals or classes of content users 112 ‘permission’ to use certain content. They may specify what kinds of content usage are permitted, and what kinds are not. They may specify how content usage is to be paid for and how much it costs. As another example, ‘rules and controls’ may require content usage information to be reported back to the distributor 106 and/or content creator 102.” (‘193 56:26-35)</p> <p>12. “ROS VDE functions 604 may be based on segmented, independently loadable executable ‘component assemblies’ 690. These component assemblies 690 are independently securely deliverable. The component assemblies 690 provided by the preferred embodiment comprise code and data elements that are themselves independently deliverable.... These component assemblies 690 are the basic</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>functional unit provided by ROS 602. The component assemblies 690 are executed to perform operating system or application tasks. Thus, some component assemblies 690 may be considered to be part of the ROS operating system 602, while other component assemblies may be considered to be 'applications' that run under the support of the operating system." ('193 83:12-29)</p> <p>13. "As mentioned above, ROS 602 provides several layers of security to ensure the security of component assemblies 690. One important security layer involves ensuring that certain component assemblies 690 are formed, loaded and executed only in secure execution space such as provided within an SPU 500." ('193 87:33-38)</p> <p>14. "Methods 1000 perform the basic function of defining what users (including, where appropriate, distributions, client administration, etc.), can and cannot do with an object 300." ('193 128:30-33)</p> <p>15. "Container 152 in this example further includes an electronic control set 188 describing conditions under which the power may be exercised. Controls 188 define the power(s) granted to each of the participants – including (in this example) conditions or limitations for exercising these powers. Controls 188 may provide the same powers and/or conditions of use for each participant, or they may provide different powers and/or conditions of use for each participant." ('712 220:1-8)</p> <p>16. "...content creators and rights owners can register permissions with the rights and permissions clearinghouses 400 in the form of electronic 'control sets.' These permissions can specify what consumers can and can't do with digital properties, under what conditions the permissions can be exercised and the consequences of exercising the permissions." ('712 72:2-7)</p> <p>17. "This 'channel 0' 'open channel' task may then issue a series of requests to secure database manager 566 to obtain the 'blueprint' for constructing one or more component assemblies 690 to be associated with channel 594 (block 1127). In the preferred embodiment, this 'blueprint' may comprise a PERC 808 and/or URT 464." ('193 112:46-51)</p> <p>(4) <u>VDE SECURE CONTAINER</u>: See support as listed for Secure Container, item #20, above.</p> <p><b>Intrinsic:</b></p> <p>1. "In part, security is enhanced by object methods employed by the present invention because the encryption schemes used to protect an object can efficiently be further used to protect the associated content control information (software control information and relevant data) from modification." ('193 15:51-55)</p> <p>2. "FIG. 5A shows how the virtual distribution environment 100, in a preferred embodiment, may package information elements (content) into a 'container' 302 so the information can't be accessed except as provided by its 'rules and controls.' Normally, the container 302 is electronic rather than physical. Electronic container 302 in one example comprises 'digital' information having a well defined structure. Container 302 and its contents can be called an 'object</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>300.” (‘193 58:39-46)</p> <ol style="list-style-type: none"> <li>3. “Moreover, when any new VDE object 300 arrives at an electronic appliance 600, the electronic appliance must ‘register’ the object within object registry 450 so that it can be accessed.” (‘193 153:56-59)</li> <li>4. “Even if the object is stored locally to the VDE node, it may be stored as a secure or protected object so that it is not directly accessible to a calling process. ACCESS method 2000 establishes the connections, routings, and security requisites needed to access the object.” (‘193 192:14-19)</li> <li>5. “ACCESS method 2000 reads the ACCESS method MDE from the secure database, reads it in accordance with the ACCESS method DTD, and loads encrypted content source and routing information based on the MDE (blocks 2010, 2012). This source and routing information specifies the location of the encrypted content. ACCESS method 2000 then determines whether a connection to the content is available (decision block 2014). This ‘connection’ could be, for example, an on-line connection to a remote site, a real-time information feed, or a path to a secure/protected resource, for example. If the connection to the content is not currently available (‘No’ exit of decision block 2014), then ACCESS method 2000 takes steps to open the connection (block 2016). If the connection fails (e.g., because the user is not authorized to access a protected secure resource), then the ACCESS method 2000 returns with a failure indication (termination point 2018).” (‘193 192:36-52)</li> <li>6. “It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g., summary, table of contents) and secured content control information which ensures the performance of control information.” (‘193 15:41-46)</li> <li>7. “In this example, creator 102 may employ one or more application software programs and one or more VDE secure subsystems to place unencrypted content into VDE protected form (i.e., into one or more VDE content containers).” (‘193 315:53-56)</li> <li>8. “The Ginter et al. patent specification referenced above describes many characteristics of this DigiBox™ container model, a powerful, flexible, general construct that enables protected, efficient and interoperable electronic description and regulation of electronic commerce relationships of all kinds...” (‘861 1:39-44)</li> <li>9. “The node and container model described above and in the Ginter et al. patent specification (along with similar other DigiBox/VDE (Virtual Distribution Environment) models) has nearly limitless flexibility.” (‘861 2:37-40)</li> <li>10. “Therefore, the container creation and usage tools must themselves be secure in the sense that they must protect certain details about the container design. This additional security requirement can make it even more difficult to make containers easy to use and to provide interoperability.” (‘861 4:59-64)</li> <li>11. “FIG. 88 illustrates secure electronic container 302 as an attaché handcuffed to the secure delivery person’s wrist. Once again, container is shown as a physical thing for purposes of illustrations only --in the example it is preferably electronic rather than physical, and comprises digital information having a well-defined structure (see FIG. 5A). Special mathematical techniques known as</li> </ol> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>'cryptography' can be used to make electronic container 302 secure so that only intended recipient 4056 can open the container and access the electronic document (or other items) 4054 it contains." ('683 15:61 - 16:14)</p> <p>12. "Appliance 600B may deliver the digital copy of item 4054 within a container 302 and/or protect the item with seals. Electronic fingerprints, watermarks and/or other visible and/or hidden markings to provide a 'virtual container' or some of the security or other characteristics of a container (for example, the ability to associate electronic controls with the item). ('683 18:49-56)</p> <p>13. "For example, defendant's attorney 5052 can specify one container 302 for opening by his co-counsel, client or client in-house counsel, and program another container 302 for opening only by opposing (plaintiff's) counsel 5050. Because of the unique trustedness features provided by system 4050, the defendant's attorney 5052 can have a high degree of trust and confidence that only the authorized parties will be able to open the respective containers and access the information they contain." ('683 56:17-25)</p> <p>14. "The 'container' concept is a convenient metaphor used to give a name to the collection of elements required to make use of content or to perform an administrative-type activity." ('193 127:30-32)</p> <p>15. "The virtual distribution environment 100, in a preferred embodiment, may package information elements (content) into a 'container' 302 so the information can't be accessed except as provided by its 'rules and controls.'" ('193 58:39-43)</p> <p>16. "VDE 100 provides a media independent container model for encapsulating content." ('193 127:2-3)</p> <p>17. "The electronic form of a document is stored as a VDE container (object) associated with the specific client and/or case. The VDE container mechanism supports a hierarchical ordering scheme for organizing files and other information with a container; this mechanism may be used to organize the electronic copies of the documents within a container. A VDE container is associated with specific access control information and rights that are described in one or more permissions control information sets (PERCs) associated with that container. In this example, only those members of the law firm who possess a VDE instance, an appropriate PERC, and the VDE object that contains the desired document, may use the document." ('193 274:52-64)</p> <p>18. "The situation is no better for processing documents within the context of ordinary computer and network systems. Although said systems can enforce access control information based on user identity, and can provide auditing mechanism for tracking accesses to files, these are low-level mechanisms that do not permit tracking or controlling the flow of content. In such systems, because document content can be freely copied and manipulated, it is not possible to determine where documents content has gone, or where it came from." ('193 281:27-35)</p> <p>19. "Secure containers 302 may be used to encapsulate the video and audio being exchanged between electronic kiosk appliances 600, 600' to maintain confidentiality and ensure a high degree of trustedness." ('682 52: 61-64)</p> <p>20. "[C]ontainer 152 can only be opened within a secure protected processing</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>environment 154 that is part of the virtual distribution environment described in the above-referenced Ginter et al. patent disclosure ...." ('712 168:22-25)</p> <p>21. "The present invention provides a new kind of 'virtual distribution environment' (called 'VDE' in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels 'across' the 'information highway.'" ('193 2:24-28)</p> <p>22. "The present invention truly achieves a content control and auditing architecture that can be configured to most any commercial distribution embodiment." ('193 261:12-15)</p> <p>23. "The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America's largest telecommunications, computer, entertainment and information provider companies to some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems." ('193 2:13-22)</p> <p>24. "The configurability provided by the present invention is particularly critical for supporting electronic commerce, that is enabling businesses to create relationships and evolve strategies that offer competitive value. Electronic commerce tools that are not inherently configurable and interoperable will ultimately fail to produce products (and services) that meet both basic requirements and evolving needs of most commerce applications." ('193 16:41-48)</p> <p>25. "VDE also extends usage control information to an arbitrary granular level (as opposed to a file based level provided by traditional operating systems) ...." ('193 275:8-11)</p> <p>26. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention: ...." ('193 21:43-45)</p> <p>27. "A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information ..." ('193 10:66 - 11:2)</p> <p>28. "Some of the key factors contributing to the configurability intrinsic to the present invention include: ...." ('193 16:66-67)</p> <p>29. "The scalable transaction management/auditing technology of the present invention will result in more efficient and reliable interoperability ...." ('193 34:9-11)</p> <p>30. "The present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components." ('193 8:63 - 9:3)</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>31. "The design of the VDE foundation, VDE load modules, and VDE containers, are important features that enable the VDE node operating environment to be compatible with a very broad range of electronic appliances." ('193 34:26-30)</p> <p>32. "The ability to optionally incorporate different methods 1000 with each object is important to making VDE 100 highly configurable." ('193 128:28-30)</p> <p>33. "An important feature of VDE is that it can be used to assure the administration of, and adequacy of security and rights protection for, electronic agreements implemented through the use of the present invention." ('193 46:51-54)</p> <p>34. "In this example, both the address request 602 and the responsive information 604 are contained within secure electronic containers 152 in order to maintain the confidentiality and integrity of the requests and responses. In this way, for example, outside eavesdroppers cannot tell who sender 95(1) wants to communicate with or what information he or she needs to perform communications with or what information he or she needs to perform the communications – and the directory responses cannot be 'spoofed' to direct the requested message to another location." ('712 12:15-22)</p> <p>35. "On the other hand, if the information to be exchanged has already been secured and/or is available without authentication (e.g., certain catalog information, containers that have already been encrypted and do not require special handling, etc.), the 'weaker' form of login/password may be used." ('193 290:57-62)</p> <p>36. "VDE provides means to securely combine content provided at different times, by differing sources, and/or representing different content types. These types, timings, and/or different sources of content can be employed to form a complex array of content within a VDE content container objects, each containing different content whose usage can be controlled, at least in part, by its own container's set of VDE content control information." ('193 297:35-45)</p> <p>37. "Although methods 1000 can have virtually unlimited variety and some may even be user-defined, certain basic 'use' type methods are preferably used in the preferred embodiment to control most of the more fundamental object manipulation and other functions provided by VDE 100. For example, the following high level methods would typically be provided for object manipulation; OPEN method, READ method, WRITE method, CLOSE method. An OPEN method is used to control opening a container so its content may be accessed. A READ method is used to control access to contents in a container. A WRITE method is used to control the insertion of contents into a container. A CLOSE method is used to close a container that has been opened." ('193 183:12-29)</p> <p>38. "DESTROY method 2180 removes the ability of a user to use an object by destroying the URT the user requires to access the object. In the preferred embodiment, .... DESTROY method 2180 may then <i>[sic]</i> call a WRITE and/or ACCESS method to write information which will corrupt (and thus destroy) the header and/or other important parts of the object (block 2186). DESTROY method 2180 may then mark one or more of the control structures (e.g., the URT) as damaged by writing appropriate information to control structure (blocks 2188, 2190)." ('193 198:41-45)</p> <p>39. "PANIC method 2200 may prevent the user from further accessing the object</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>currently being accessed by, for example, destroying the channel being used to access the object and marking one or more of the control structures (e.g., the URT) associated with the user and object as damaged.(blocks 2206, and 2208-2210, respectively). Because the control structure is damaged, the VDE node will need to contact an administrator to obtain a valid control structure(s) before the user may access the same object again.” (‘193 198:60 - 199:2)</p> <p>40. “EXTRACT method 2080 is used to copy or remove content from an object and place it into a new object. In the preferred embodiment, the EXTRACT method 2080 does not involve any release of content, but rather simply takes content from one container and places it into another container, both of which may be secure. Extraction of content differs from release in that the content is never exposed outside a secure container.” (‘193 194:13-20)</p> <p>41. “Use of secure electronic containers to transport items provides an unprecedented degree of security, trustedness and flexibility.” (‘683 8:50-52)</p> <p>42. “Electronic delivery person 4060 can deliver the electronic version of item 4054 within secure container attaché case 302 from personal computer 4116’ to another personal computer 4116 operated by recipient 4056.” (‘683 20:27-30)</p> <p>43. “Because these transactions are conducted using VDE and VDE secure containers, those observing the communications learn no more than the fact that the parties are communicating.” (‘712 310:1-3)</p> <p>44. “VDE in one example provides a ‘virtual silicon container’ (‘virtual black box’) in that several different instances of SPU 500 may securely communicate together to provide an overall secure hardware environment that ‘virtually’ exists at multiple locations and multiple electronic appliances 600. FIG. 87 shows one model 3600 of a virtual silicon container. This virtual container model 3600 includes a content creator 102, a content distributor 106, one or more content redistributors 106a, one or more client administrators 700, one or more client users 3602, and one or more clearinghouses 116. Each of these various VDE participants has an electronic appliance 600 including a protected processing environment 655 that may comprise, at least in part, a silicon-based semiconductor hardware element secure processing unit 500. The various SOUs 500 each encapsulate a part of the virtual distribution environment, and thus, together form the virtual silicon container 3600.” (‘193 317:58 - 318:8)</p> <p>45. “Uses tools to transform digital information(such as electronic books, databases, computer software and movies) into protected digital packages called ‘objects.’ Only those consumers (or other along the chain of possession such as redistributor) who receive permission from a distributor 106 can open these packages. VDE packaged content can be constrained by ‘rules and control information.’” (‘193 254:18-25)</p> <p>46. “To open VDE package and make use of its content, and end-user must have permission.” (‘193 254:45-46)</p> <p>47. “Place unencrypted content into VDE protected form (i.e., into one or more VDE content containers).” (‘193 315:55-56)</p> <p>(5) <u>NON-CIRCUMVENTABLE:</u><br/>Intrinsic:</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <ol style="list-style-type: none"> <li>1. "VDE can protect a collection of rights belonging to various parties having in rights in, or to, electronic information. This information may be at one location or dispersed across (and/or moving between) multiple locations. The information may pass through a 'chain' of distributors and a 'chain' of users. Usage information may also be reported through one or more 'chains' of parties. In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed." ('193 6:18-31)</li> <li>2. "All requirements specified by this derived control information must be satisfied before VDE controlled content can be accessed or otherwise used." ('193 11:8-11)</li> <li>3. "VDE provides important mechanisms for both enforcing commercial agreements and enabling the protection of privacy rights. VDE can securely deliver information from one party to another concerning the use of commercially distributed electronic content. Even if parties are separated by several 'steps' in a chain (pathway) of handling for such content usage information, such information is protected by VDE through encryption and/or other secure processing. Because of that protection, the accuracy of such information is guaranteed by VDE, and the information can be trusted by all parties to whom it is delivered." ('193 14:29-39)</li> <li>4. "VDE ensures that certain prerequisites necessary for a given transaction to occur are met. This includes the secure execution of any required load modules and the availability of any required, associated data." ('193 20:27-30)</li> <li>5. "Required methods (methods listed as required for property and/or appliance use) must be available as specified if VDE controlled content (such as intellectual property distributed within a VDE content container) is to be used." ('193 43:37-41)</li> <li>6. "Since all secure communications are at least in part encrypted and the processing inside the secure subsystem is concealed from outside observation and interference, the present invention ensures that content control information can be enforced." ('193 46:4-8)</li> <li>7. "This control information can determine, for example: <ol style="list-style-type: none"> <li>(1) How and/or to whom electronic content can be provided, for example, how an electronic property can be distributed;</li> <li>(2) How one or more objects and/or properties, or portions of an object or property, can be directly used, such as decrypted, displayed, printed, etc; ...." ('193 46:17-24)</li> </ol> </li> <li>8. "'Hardware' 506 also contains long-term and short-term memories to store information securely so it can't be tampered with." ('193 60:1-3)</li> <li>9. "A feature of VDE provided by the present invention is that certain one or more methods can be specified as required in order for a VDE installation and/or user to be able to use certain and/or all content." ('193 43:47-50)</li> <li>10. "The virtual distribution environment 100 prevents use of protected information</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>except as permitted by the 'rules and controls' (control information)." ('193 56:26-28)</p> <p>11. "As mentioned above, virtual distribution environment 100 'associates' content with corresponding 'rules and controls,' and prevents the content from being used or accessed unless a set of corresponding 'rules and controls' is available. The distributor 106 doesn't need to deliver content to control the content's distribution. The preferred embodiment can securely protect content by protecting corresponding, usage enabling 'rules and controls' against unauthorized distribution and use." ('193 57:18-26)</p> <p>12. "Since no one can use or access protected content without 'permission' from corresponding 'rules and controls,' the distributor 106 can control use of content that has already been (or will in the future be) delivered." ('193 57:30-33)</p> <p>13. "SPU 500 is enclosed within and protected by a 'tamper resistant security barrier' 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions. Barrier 502 also controls external access to secure resources, processes and information within SPU 500." ('193 59:48-55)</p> <p>14. "Provides non-repudiation of use and may record specific forms of use such as viewing, editing, extracting, copying, redistributing (including to what one or more parties), and/or saving." ('683 6:46-48)</p> <p>15. "In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed." ('193 6:24-30)</p> <p>16. "To securely control access and other use, including distribution of records, documents, and notes associated with the case" ('193 274:34-36)</p> <p>17. "Thus wrapped, a VDE object may be distributed to the recipient without fear of unauthorized access and/or other use." ('193 277:16-17)</p> <p>18. "These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc." ('193 9:24-27)</p> <p>19. "VDE provides a secure, distributed electronic transaction management system for controlling the distribution and/or other usage of electronically provided and/or stored information." ('193 9:36-39)</p> <p>20. "The control set 404 might permit publisher 168 to add his own additional controls that allow consumer 95 to read the work 166 an unlimited number of time but prevent the consumer from copying or redistributing the work." (712 258: 8-11)</p> <p>21. "The doctor 5000 may then send container 301(1) to a trusted go-between 4700. ... For example, the trusted go-between 4700 in one example has no access to the content of the container 302(1), but does have a record of a seal of</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>the contents.” (‘683 53:40-57)</p> <p>22. “FIG. 116 shows example steps that may be performed by PPE 650 in response to an ‘open’ or ‘view’ event. In this example, PPE 650 may - - upon allowing recipient 4056 to actually interact with the item 4054—...PPE 650 may then release the image 4068I and/or the data 4068D to the application running on electronic appliance 600—electronic fingerprinting or watermarking the released content if appropriate (FIG. 116, block 4625C). (‘683 42:38-52)</p> <p>23. “FIG. 5A shows how the virtual distribution environment 100, in a preferred embodiment, may package information elements (content) into a ‘container’ 302 so the information can’t be accessed except as provided by its ‘rules and controls.’” (‘193 58:39-43)</p> <p><b>(6) <u>PEER TO PEER:</u></b></p> <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “Each VDE participant in a VDE pathway of content control information may set methods for some or all of the content in a VDE container, so long as such control information does not conflict with senior control information already in place with respect to: <ol style="list-style-type: none"> <li>(1) certain or all VDE managed content,</li> <li>(2) certain one or more VDE users and/or groupings of users,</li> <li>(3) certain one or more VDE nodes and/or groupings of nodes, and/or</li> <li>(4) certain one or more VDE applications and/or arrangements.” (‘193 44:6-17)</li> </ol> </li> <li>2. “All participants of VDE 100 have the innate ability to participate in any role.” (‘193 256:50-51)</li> <li>3. “Any VDE user 112 may assign the right to process information or perform services on their behalf to the extent allowed by senior control information.” (‘193 257:17-20)</li> <li>4. “PERC and URT structures provide a mechanism that may be used to provide precise electronic representation of rights and the controls associated with those rights. VDE thus provides a ‘vocabulary’ and mechanism by which users and creators may specify their desires.” (‘193 245:11-15)</li> </ol> <p><b>(7) <u>COMPREHENSIVE RANGE OF FUNCTIONS:</u></b></p> <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “VDE provides comprehensive and configurable transaction management, metering and monitoring technology.” (‘193 3:34-35)</li> <li>2. “VDE may be combined with, or integrated into, many separate computers and/or other electronic appliances. These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred embodiment comprises one or more ‘protected processing environments’, one or more secure databases, and secure ‘component assemblies’ and other items and processes that need to be kept secured. VDE can, for example, securely control electronic currency, payments, and/or credit management (including electronic credit and/or currency receipt, disbursement, encumbering, and/or allocation) using such a ‘secure subsystem.’” (‘193 9:22-35)</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>3. "In addition VDE:</p> <ul style="list-style-type: none"> <li>(a) is very configurable, modifiable, and re-usable;</li> <li>(b) supports a wide range of useful capabilities that may be combined in different ways to accommodate most potential applications;</li> <li>(c) operates on a wide variety of electronic appliances ranging from hand-held inexpensive devices to large mainframe computers;</li> <li>(d) is able to ensure the various rights of a number of different parties, and a number of different rights protection schemes, simultaneously;</li> <li>(e) is able to preserve the rights of parties through a series of transactions that may occur at different times and different locations;</li> <li>(f) is able to flexibly accommodate different ways of securely delivering information and reporting usage; and</li> <li>(g) provides for electronic analogues to 'real' money and credit, including anonymous electronic cash, to pay for products and services and to support personal (including home) banking and other financial activities." ('193 4:57 - 5:10)</li> </ul> <p>4. "[VDE] can provide efficient, reusable, modifiable, and consistent means for secure electronic content: distribution, usage control, usage payment, usage auditing, and usage reporting." ('193 8:26-29)</p> <p>5. "VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment." ('193 8:53-58)</p> <p>6. "The present invention allows content providers and users to formulate their transaction environment to accommodate:</p> <ul style="list-style-type: none"> <li>(1) desired content models, content control models, and content usage information pathways,</li> <li>(2) a complete range of electronic media and distribution means,</li> <li>(3) a broad range of pricing, payment, and auditing strategies,</li> <li>(4) very flexible privacy and/or reporting models,</li> <li>(5) practical and effective security architectures, and</li> <li>(6) other administrative procedures that together with steps (1) through (5) can enable most 'real world' electronic commerce and data security models, including models unique to the electronic world." ('193 10:11-23)</li> </ul> <p>7. "Because of the breadth of issues resolved by the present invention, it can provide the emerging 'electronic highway' with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions." ('193 17:22-28)</p> <p>8. "A feature of the present invention provides for payment means supporting flexible electronic currency and credit mechanisms, including the ability to securely maintain audit trails reflecting information related to use of such currency or credit." ('193 33:58-63)</p> <p>9. "The end-to-end nature of VDE applications, in which content 108 flows in one</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>direction, generating reports and bills 118 in the other, makes it possible to perform 'back-end' consistency checks." ('193 223:17-20)</p> <p>10. "By way of non-exhaustive summary, these present inventions provide a highly secure and trusted item delivery and agreement execution services providing the following features and functions:</p> <p>Trustedness and security approaching or exceeding that of a personal trusted courier.</p> <p>Instant or nearly instant delivery.</p> <p>Optional delayed delivery ("store and forward").</p> <p>Broadcasting to multiple parties.</p> <p>Highly cost effective.</p> <p>Trusted validation of item contents and delivery.</p> <p>Value Added Delivery and other features selectable by the sender and/or recipient.</p> <p>Provides electronic transmission trusted auditing and validating.</p> <p>Allows people to communicate quickly, securely, and confidentially.</p> <p>Communications can later be proved through reliable evidence of the communications transaction--providing non-repudiatable, certain, admissible proof that a particular communications transaction occurred.</p> <p>Provides non-repudiation of use and may record specific forms of use such as viewing, editing, extracting, copying, redistributing (including to what one or more parties), and/or saving.</p> <p>Supports persistent rights and rules based document workflow management at recipient sites.</p> <p>System may operate on the Internet, on internal organization and/or corporate networks ("intranets" irrespective of whether they use or offer Internet services internally), private data networks and/or using any other form of electronic communications.</p> <p>System may operate in non-networked and/or intermittently networked environments.</p> <p>Legal contract execution can be performed in real time, with or without face to face or ear-to-ear personal interactions (such as audiovisual teleconferencing, automated electronic negotiations, or any combination of such interactions) for any number of distributed individuals and/or organizations using any mixture of interactions.</p> <p>The items delivered and/or processed may be any 'object' in digital format, including, but not limited to, objects containing or representing data types such as text, images, video, linear motion pictures in digital format, sound recordings and other audio information, computer software, smart agents, multimedia, and/or objects any combination of two or more data types contained within or representing a single compound object.</p> <p>Content (executables for example) delivered with proof of delivery and/or execution or other use.</p> <p>Secure electronic containers can be delivered. The containers can maintain control, audit, receipt and other information and protection securely and persistently in association with one or more items.</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>Trustedness provides non-repudiation for legal and other transactions.</p> <p>Can handle and send any digital information (for example, analog or digital information representing text, graphics, movies, animation, images, video, digital linear motion pictures, sound and sound recordings, still images, software computer programs or program fragments, executables, data, and including multiple, independent pieces of text; sound clips, software for interpreting and presenting other elements of content, and anything else that is electronically representable).</p> <p>Provides automatic electronic mechanisms that associate transactions automatically with other transactions.</p> <p>System can automatically insert or embed a variety of visible or invisible 'signatures' such as images of handwritten signatures, seals, and electronic 'fingerprints' indicating who has 'touched' (used or other interacted with in any monitorable manner) the item.</p> <p>System can affix visible seals on printed items such as documents for use both in encoding receipt and other receipt and/or usage related information and for establishing a visible presence and impact regarding the authenticity, and ease of checking the authenticity, of the item.</p> <p>Seals can indicate who originated, sent, received, previously received and redistributed, electronically view, and/or printed and/or otherwise used the item.</p> <p>Seals can encode digital signatures and validation information providing time, location, send and/or other information and/or providing means for item authentication and integrity check.</p> <p>Scanning and decoding of item seals can provide authenticity/integrity check of entire item(s) or part of an item (e.g., based on number of words, format, layout, image--picture and/or test--composition, etc.).</p> <p>Seals can be used to automatically associate electronic control sets for use in further item handling.</p> <p>System can hide additional information within the item using 'steganography' for later retrieval and analysis.</p> <p>Steganography can be used to encode electronic fingerprints and/or other information into an item to prevent deletion.</p> <p>Multiple steganographic storage of the same fingerprint information may be employed reflecting 'more' public and 'less' public modes so that a less restricted steganographic mode (different encryption algorithm, keys, and/or embedding techniques) can be used to assist easy recognition by an authorized party and a more private (confidential) mode may be readable by only a few parties (or only one party) and comprise of the less restricted mode may not affect the security of the more private mode.</p> <p>Items such as documents can be electronically, optically scanned at the sender's end--and printed out in original, printed form at the recipient's end.</p> <p>Document handlers and processors can integrate document scanning and delivery.</p> <p>Can be directly integrated into enterprise and Internet (and similar network) wide document workflow systems and applications.</p> <p>Secure, tamper-resistant electronic appliance, which may employ VDE SPUs,</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>used to handle items at both sender and recipient ends.</p> <p>'Original' item(s) can automatically be destroyed at the sender's end and reconstituted at the recipient's end to prevent two originals from existing simultaneously.</p> <p>Secure, non-repudiable authentication of the identification of a recipient before delivery using any number of different authentication techniques including but not limited to biometric techniques (such as palm print scan, signature scan, voice scan, retina scan, iris scan, biometric fingerprint and/or handprint scan, and/or face profile) and/or presentation of a secure identity 'token.'</p> <p>Non-repudiation provided through secure authentication used to condition events (e.g., a signature is affixed onto a document only if the system securely authenticates the sender and her intention to agree to its contents).</p> <p>Variety of return receipt options including but not limited to a receipt indicating who opened a document, when, where, and the disposition of the document (stored, redistributed, copied, etc.). These receipts can later be used in legal proceedings and/or other contexts to prove item delivery, receipt and/or knowledge.</p> <p>Audit, receipt, and other information can be delivered independently from item delivery, and become securely associated with an item within a protected processing environment.</p> <p>Secure electronic controls can specify how an item is to be processed or otherwise handled (e.g., document can't be modified, can be distributed only to specified persons, collections of persons, organizations, can be edited only by certain persons and/or in certain manners, can only be viewed and will be 'destroyed' after a certain elapse of time or real time or after a certain number of handlings, etc.)</p> <p>Persistent secure electronic controls can continue to supervise item workflow even after it has been received and 'read.'</p> <p>Use of secure electronic containers to transport items provides an unprecedented degree of security, trustedness and flexibility.</p> <p>Secure controls can be used in conjunction with digital electronic certificates certifying as to identity, class (age, organization membership, jurisdiction, etc.) of the sender and/or receiver and/or user of communicated information.</p> <p>Efficiently handles payment and electronic addressing arrangements through use of support and administrative services such as a Distributed Commerce Utility as more fully described in the copending Shear, et al. application.</p> <p>Compatible with use of smart cards, including, for example, VDE enabled smart cards, for secure personal identification and/or for payment.</p> <p>Transactions may be one or more component transactions of any distributed chain of handling and control process including Electronic Data Interchange (EDI) system, electronic trading system, document workflow sequence, and banking and other financial communication sequences, etc." ('683 6:18 - 9:4)</p> <p>11. "Content providers and distributors have devised a number of limited function rights protection mechanisms to protect their rights. Authorization passwords and protocols, license servers, 'lock/unlock' distribution methods, and non-electronic contractual limitations imposed on users of shrink-wrapped software</p> |

| Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|----------------------|---|
|                      | <p>are a few of the more prevalent content protection schemes. In a commercial context, these efforts are inefficient and limited solutions.” (‘193 3:1-9)</p> <p>(8) <u>USER-CONFIGURABLE:</u><br/> <b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America’s largest telecommunications, computer, entertainment and information provider companies to some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems.” (‘193 2:13-22)</li> <li>2. “The features of VDE allow it to function as the first trusted electronic information control environment that can conform to, and support, the bulk of conventional electronic commerce and data security requirements. In particular, VDE enables the participants in a business value chain model to create an electronic version of traditional business agreement terms and conditions and further enables these participants to shape and evolve their electronic commerce models as they believe appropriate to their business requirements.” (‘193 8:43-52)</li> <li>3. “An objective of VDE is supporting a transaction/distribution control standard. Development of such a standard has many obstacles, given the security requirements and related hardware and communications issues, widely differing environments, information types, types of information usage, business and/or data security goals, varieties of participants, and properties of delivered information. A significant feature of VDE accommodates the many, varying distribution and other transaction variables by, in part, decomposing electronic commerce and data security functions into generalized capability modules executable within a secure hardware SPU and/or corresponding software subsystem and further allowing extensive flexibility in assembling, modifying, and/or replacing, such modules (e.g. load modules and/or methods) in applications run on a VDE installation foundation. This configurability and reconfigurability allows electronic commerce and data security participants to reflect their priorities and requirements through a process of iteratively shaping an evolving extended electronic agreement (electronic control model).” (‘193 15:66 - 16:18)</li> <li>4. “Some of the key factors contributing to the configurability intrinsic to the present invention include:               <ol style="list-style-type: none"> <li>(a) integration into the fundamental control environment of a broad range of electronic appliances through portable API and programming language tools that efficiently support merging of control and auditing capabilities in nearly any electronic appliance environment while maintaining overall system security;</li> <li>(b) modular data structures;</li> <li>(c) generic content model;</li> </ol> </li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>(d) general modularity and independence of foundation architectural components;</p> <p>(e) modular security structures;</p> <p>(f) variable length and multiple branching chains of control; and</p> <p>(g) independent, modular control structures in the form of executable load modules that can be maintained in one or more libraries, and assembled into control methods and models, and where such model control schemes can 'evolve' as control information passes through the VDE installations of participants of a pathway of VDE content control information handling." ('193 16:66 - 17:21)</p> <p>5. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention: ... VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, ... provide mechanisms that allow control information to 'evolve' and be modified according, at least in part, to independently, securely delivered further control information.... Handlers in a pathway of handling of content control information, to the extent each is authorized, can establish, modify, and/or contribute to, permission, auditing, payment, and reporting control information related to controlling, analyzing, paying for, and/or reporting usage of, electronic content and/or appliances (for example, as related to usage of VDE controlled property content)." ('193 21:43-46; 29:21-41)</p> <p>6. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention: ... VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, ... enable a user to securely extract, through the use of the secure subsystem at the user's VDE installation, at least a portion of the content included within a VDE content container to produce a new, secure object (content container), such that the extracted information is maintained in a continually secure manner through the extraction process." ('193 21:43-46; 31:66 - 32:5)</p> <p>7. "As with the content control information for most VDE managed content, features of the present invention allows [sic] the content's control information to: (a) 'evolve,' for example, the extractor of content may add new control methods and/or modify control parameter data, such as VDE application compliant methods, to the extent allowed by the content's in-place control information. ... (b) allow a user to combine additional content with at least a portion of said extracted content, ... (c) allow a user to securely edit at least a portion of said content while maintaining said content in a secure form within said VDE content container; ... (d) append extracted content to a pre-existing VDE content container object and attach associated control information ... (e) preserve VDE control over one or more portions of extracted content after</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>various forms of usage of said portions ... Generally, the extraction features of the present invention allow users to aggregate and/or disseminate and/or otherwise use protected electronic content information extracted from content container sources while maintaining secure VDE capabilities thus preserving the rights of providers in said content information after various content usage processes." ('193 32:27 - 33:4)</p> <p>8. "The secure component based architecture of ROS 602 has important advantages. For example, it accommodates limited resource execution environments such as provided by a lower cost SPU 500. It also provides an extremely high level of configurability. In fact, ROS 602 will accommodate an almost unlimited diversity of content types, content provider objectives, transaction types and client requirements. In addition, the ability to dynamically assemble independently deliverable components at execution time based on particular objects and users provides a high degree of flexibility" ('193 87:63 - 88:7)</p> <p>9. "Each logical object structure 800 may also include a 'private body' 806 containing or referencing a set of methods 1000 (i.e., programs or procedures) that control use and distribution of the object 300. The ability to optionally incorporate different methods 1000 with each object is important to making VDE 100 highly configurable." ('193 128:25-30)</p> <p>10. "VDE methods 1000 are designed to provide a very flexible and highly modular approach to secure processing." ('193 181:17-18)</p> <p>11. "The reusable functional primitives of VDE 100 can be flexibly combined by content providers to reflect their respective distribution objectives." ('193 255:27-29)</p> <p>12. "The present invention truly achieves a content control and auditing architecture that can be configured to most any commercial distribution embodiment." ('193 261:12-15)</p> <p>13. "Adding new content to objects is an important aspect of authoring provided by the present invention. Providers may wish to allow one or more users to add, hide, modify, remove and/or extend content that they provide. In this way, other users may add value to, alter for a new purpose, maintain, and/or otherwise change, existing content. The ability to add content to an empty and/or newly created object is important as well." ('193 261:23-30)</p> <p>14. "The distribution control information provided by the present invention allows flexible positive control. No provider is required to include any particular control, or use any particular strategy, except as required by senior control information. Rather, the present invention allows a provider to select from generic control components (which may be provided as a subset of components appropriate to a provider's specific market, for example, as included in and/or directly compatible with, a VDE application) to establish a structure appropriate for a given chain of handling/control." ('193 263:9-19)"</p> <p>15. "Importantly, VDE securely and flexibly supports editing the content in, extracting content from, embedding content into, and otherwise shaping the content composition of, VDE content containers. Such capabilities allow VDE supported product models to evolve by progressively reflecting the</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>requirements of 'next' participants in an electronic commercial model." ('193 297:9-15)</p> <p>16. "For instance, the user may have an 'access' right, and an 'extraction' right, but not a 'copy' right." ('193 159:24-26)</p> <p>17. "PERCS 808 specify a set of rights that may be exercised to use or access the corresponding VDE object 300. The preferred embodiment allows users to 'customize' their access rights by selecting a subset of rights authorized by a corresponding PERC 808 and/or by specifying parameters or choices that correspond to some or all of the rights granted by PERC 808. These user choices are set forth in a user rights table 464 in the preferred embodiment. User rights table (URT) 464 includes URT records, each of which correspond to a user (or group of users). Each of these URT records specific users choices for a corresponding VDE object more methods 1000 for exercising the rights granted to the user by the PERC 808 in a way specified by the choices contained within the URT record." ('193 156:55 - 157:3)</p> <p>18. "PERC and URT structures provide a mechanism that may be used to provide precise electronic representation of rights and the controls associated with those rights. VDE thus provides a 'vocabulary' and mechanism by which users and creators may specify their desires." ('193 245:10-15)</p> <p>19. "In sum, the present invention allows information contained in electronic information products to be supplied according to user specification. Tailoring to user specification allows the present invention to provide the greatest value to users, which in turn will generate the greatest amount of electronic commerce activity." ('193 22:66 - 23:5)</p> <p>20. "Adding new content to objects is an important aspect of authoring provided by the present invention. Providers may wish to allow one or more users to add, hide, modify, remove and/or extend content that they provide. In this way, other users may add value to, alter for a new purpose, maintain, and/or otherwise change, existing content. The ability to add content to an empty and/or newly created object is important as well." ('193 261:23-30)</p> <p>21. "Each logical object structure 800 may also include a 'private body' 806 containing or referencing a set of method 1000 (i.e., programs or procedures) that control use and distribution of the object 300. The ability to optionally incorporate different methods 1000 with each object is important to making VDE 100 highly configurable." ('193 128:25-30)</p> <p>22. "An important aspect of adding or modifying content is the choice of encryption/decryption keys and/or other relevant aspects of securing new or altered content." ('193 262:21-23)</p> <p><b>(9) GENERAL PURPOSE; UNIVERSAL:</b></p> <p><b>Intrinsic:</b></p> <p>1. "VDE also features fundamentally important capabilities for managing content that travels 'across' the 'information highway.' These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users, and others. VDE is the first general purpose,</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.” (‘193 2:27-36)</p> <p>2. “VDE provides a unified solution that allows all content creators, providers, and users to employ the same electronic rights protection solution.” (‘193 5:17-19)</p> <p>3. “Since different groups of components can be put together for different applications, the present invention can provide electronic control information for a wide variety of different products and markets. This means the present invention can provide a ‘unified,’ efficient, secure, and cost-effective system for electronic commerce and data security. This allows VDE to serve as a single standard for electronic rights protection, data security, and electronic currency and banking.” (‘193 7:6-14)</p> <p>4. “Employing VDE as a general purpose electronic transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model. For example, VDE allows content creators to use the same VDE foundation control arrangement for both content authoring and for licensing content from other content creators for inclusion into their products or for other use. Clearinghouses, distributors, content creators, and other VDE users can all interact, both with the applications running on their VDE installations, and with each other, in an entirely consistent manner, using and reusing (largely transparently) the same distributed tools, mechanisms, and consistent user interfaces, regardless of the type of VDE activity.” (‘193 11:38-59)</p> <p>5. “An objective of VDE is supporting a transaction/distribution control standard.” (‘193 15:66-67)</p> <p>6. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... The design of the VDE foundation, VDE load modules, and VDE containers, are important features that enable the VDE node operating environment to be compatible with a very broad range of electronic appliances. The ability, for example, for control methods based on load modules to execute in very ‘small’ and inexpensive secure sub-system environments, such as environments with very little read/write memory, while also being able to execute in large memory sub-systems that may be used in more expensive electronic appliances, supports consistency across many machines. This consistent VDE operating environment, including its control structures and container architecture, enables the use of standardized VDE content containers across a broad range of device types and host operating environments. Since VDE capabilities can be seamlessly integrated as extensions, additions, and/or modifications to fundamental capabilities of</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>electronic appliances and host operating systems, VDE containers, content control information, and the VDE foundation will be able to work with many device types and these device types will be able to consistently and efficiently interpret and enforce VDE control information." ('193 21:43-46; 34:26-49)</p> <p>7. "This rationalization stems from the reusability of control structures and user interfaces for a wide variety of transaction management related activities. As a result, content usage control, data security, information auditing, and electronic financial activities, can be supported with tools that are reusable, convenient, consistent, and familiar. In addition, a rational approach--a transaction/distribution control standard--allows all participants in VDE the same foundation set of hardware control and security, authoring, administration, and management tools to support widely varying types of information, business market model, and/or personal objectives." ('193 11:26-37)</p> <p>8. "Because of the breadth of issues resolved by the present invention, it can provide the emerging 'electronic highway' with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions. VDE's electronic transaction management mechanisms can enforce the electronic rights and agreements of all parties participating in widely varying business and data security models, and this can be efficiently achieved through a single VDE implementation within each VDE participant's electronic appliance. VDE supports widely varying business and/or data security models that can involve a broad range of participants at various 'levels' of VDE content and/or content control information pathways of handling. Different content control and/or auditing models and agreements may be available on the same VDE installation. These models and agreements may control content in relationship to, for example, VDE installations and/or users in general; certain specific users, installations, classes and/or other groupings of installations and/or users; as well as to electronic content generally on a given installation, to specific properties, property portions, classes and/or other groupings of content." ('193 17:22-45)</p> <p>9. "The present invention's trusted/secure, universe wide, distributed transaction control and administration system." ('193 35:66 - 36:1)</p> <p>10. "Commerce Utility Systems 90 are generalized and programmable..." ('712 67:7-8)</p> <p>(10) <u>FLEXIBLE:</u></p> <p><b>Intrinsic:</b></p> <p>1. "Providers of 'electronic currency' have also created protections for their type of content. These systems are not sufficiently adaptable, efficient, nor flexible enough to support the generalized use of electronic currency. Furthermore, they do not provide sophisticated auditing and control configuration capabilities. This means that current electronic currency tools lack the sophistication needed</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>for many real-world financial business models. VDE provides means for anonymous currency and for 'conditionally' anonymous currency, wherein currency related activities remain anonymous except under special circumstances." ('193 3:10-20)</p> <ol style="list-style-type: none"> <li>2. "Traditional content control mechanisms often require users to purchase more electronic information than the user needs or desires. For example, infrequent users of shrink-wrapped software are required to purchase a program at the same price as frequent users, even though they may receive much less value from their less frequent use. Traditional systems do not scale cost according to the extent or character of usage and traditional systems can not attract potential customers who find that a fixed price is too high. Systems using traditional mechanisms are also not normally particularly secure. For example, shrink-wrapping does not prevent the constant illegal pirating of software once removed from either its physical or electronic package." ('193 5:50-62)</li> <li>3. "Traditional electronic information rights protection systems are often inflexible and inefficient and may cause a content provider to choose costly distribution channels that increase a product's price. In general these mechanisms restrict product pricing, configuration, and marketing flexibility. These compromises are the result of techniques for controlling information which cannot accommodate both different content models and content models which reflect the many, varied requirements, such as content delivery strategies, of the model participants. This can limit a provider's ability to deliver sufficient overall value to justify a given product's cost in the eyes of many potential users. VDE allows content providers and distributors to create applications and distribution networks that reflect content providers' and users' preferred business models. It offers users a uniquely cost effective and feature rich system that supports the ways providers want to distribute information and the ways users want to use such information." ('193 5:63 - 6:13)</li> <li>4. "VDE does not require electronic content providers and users to modify their business practices and personal preferences to conform to a metering and control application program that supports limited, largely fixed functionality. Furthermore, VDE permits participants to develop business models not feasible with non- electronic commerce, for example, involving detailed reporting of content usage information, large numbers of distinct transactions at hitherto infeasible low price points, 'pass-along' control information that is enforced without involvement or advance knowledge of the participants, etc." ('193 9:67 - 10:9)</li> <li>5. "VDE can further be used to enable commercially provided electronic content to be made available to users in user defined portions, rather than constraining the user to use portions of content that were 'predetermined' by a content creator and/or other provider for billing purposes." ('193 11:66 - 12:4)</li> <li>6. "The 'usage map' concept provided by the preferred embodiment may be tied to the concept of 'atomic elements.' In the preferred embodiment, usage of an object 300 may be metered in terms of 'atomic elements.' In the preferred embodiment, an 'atomic element' in the metering context defines a unit of usage that is 'sufficiently significant' to be recorded in a meter. The definition</li> </ol> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>of what constitutes an 'atomic element' is determined by the creator of an object 300. For instance, a 'byte' of information content contained in an object 300 could be defined as an 'atomic element,' or a record of a database could be defined as an 'atomic element,' or each chapter of an electronically published book could be defined as an 'atomic element.'" ('193 144:53-65)</p> <p>7. "Summary of Some Important Features Provided by VDE in Accordance With the Present Invention. VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, VDE includes features that. . . support dynamic user selection of information subsets of a VDE electronic information product (VDE controlled content). This contrasts with the constraints of having to use a few high level individual, pre-defined content provider information increments such as being required to select a whole information product or product section in order to acquire or otherwise use a portion of such product or section. VDE supports metering and usage control over a variety of increments (including 'atomic' increments, and combinations of different increment types) that are selected ad hoc by a user and represent a collection of pre-identified one or more increments (such as one or more blocks of a preidentified nature, e.g., bytes, images, logically related blocks) that form a generally arbitrary, but logical to a user, content 'deliverable.' VDE control information (including budgeting, pricing and metering) can be configured so that it can specifically apply, as appropriate, to ad hoc selection of different, unanticipated variable user selected aggregations of information increments and pricing levels can be, at least in part, based on quantities and/or nature of mixed increment selections (for example, a certain quantity of certain text could mean associated images might be discounted by 15%; a greater quantity of text in the 'mixed' increment selection might mean the images are discounted 20%). Such user selected aggregated information increments can reflect the actual requirements of a user for information and is more flexible than being limited to a single, or a few, high level, (e.g. product, document, database record) predetermined increments. Such high level increments may include quantities of information not desired by the user and as a result be more costly than the subset of information needed by the user if such a subset was available. In sum, the present invention allows information contained in electronic information products to be supplied according to user specification. Tailoring to user specification allows the present invention to provide the greatest value to users, which in turn will generate the greatest amount of electronic commerce activity. The user, for example, would be able to define an aggregation of content derived from various portions of an available content product, but which, as a deliverable for use by the user, is an entirely unique aggregated increment. The user may, for example, select certain numbers of bytes of information from various portions of an information product, such as a reference work, and copy them to disc in unencrypted form and be billed based on total number of bytes plus a surcharge on the number of 'articles' that provided the bytes. A content provider might</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>reasonably charge less for such a user defined information increment since the user does not require all of the content from all of the articles that contained desired information.” (‘193 21:43-53; 22:32-49)</p> <p>8. “Summary of Some Important Features Provided by VDE in Accordance With the Present Invention ... Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments.” (‘193 21:43-46; 28:23-28)</p> <p>9. “The VDE templates, classes, and control structures are inherently flexible and configurable to reflect the breadth of information distribution and secure storage requirements, to allow for efficient adaptation into new industries as they evolve, and to reflect the evolution and/or change of an existing industry and/or business, as well as to support one or more groups of users who may be associated with certain permissions and/or budgets and object types. The flexibility of VDE templates, classes, and basic control structures is enhanced through the use of VDE aggregate and control methods which have a compound, conditional process impact on object control. Taken together, and employed at times with VDE administrative objects and VDE security arrangements and processes, the present invention truly achieves a content control and auditing architecture that can be configured to most any commercial distribution embodiment. Thus, the present invention fully supports the requirements and biases of content providers without forcing them to fit a predefined application model. It allows them to define the rights, control information, and flow of their content (and the return of audit information) through distribution channels.” (‘193 260:66 - 261:20)</p> <p>10. “VDE also extends usage control information to an arbitrary granular level (as opposed to a file based level provided by traditional operating systems) and provides flexible control information over any action associated with the information which can be described as a VDE controlled process.” (‘193 275:8-13)</p> <p>11. “The situation is no better for processing documents within the context of ordinary computer and network systems. Although said systems can enforce access control information based on user identity, and can provide auditing mechanisms for tracking accesses to files, these are low-level mechanisms that do not permit tracking or controlling the flow of content. In such systems, because document content can be freely copied and manipulated, it is not possible to determine where document content has gone, or where it came from. In addition, because the control mechanisms in ordinary computer operating systems operate at a low level of abstraction, the entities they control are not necessarily the same as those that are manipulated by users. This particularly causes audit trails to be cluttered with voluminous information describing uninteresting activities.” (‘193 281:27-41)</p> <p>12. “Importantly, VDE securely and flexibly supports editing the content in, extracting content from, embedding content into, and otherwise shaping the content composition of, VDE content containers.” (‘193 297:9-12)</p> |

|     | Claim Term/Phrase  | Evidence Supporting MS Construction   |
|-----|--|---|
|     |  | <p>13. "The InterTrust DigiBox container model allows and facilitates these and other different container uses. It facilitates detailed container customization for different uses, classes of use and/or users in order to meet different needs and business models. This customization ability is very important, particularly when used in conjunction with a general purpose, distributed rights management environment such as described in Ginter, et al. Such an environment calls for a practical optimization of customizability, including customizability and transparency for container models. This customization flexibility has a number of advantages, such as allowing optimization (e.g., maximum efficiency, minimum overhead) of the detailed container design for each particular application or circumstance so as to allow many different container designs for many different purposes (e.g., business models) to exist at the same time and be used by the rights control client (node) on a user electronic appliance such as a computer or entertainment device." ('861 2:49-67)</p> <p>14. "The node and container model described above and in the Ginter et al. patent specification (along with similar other DigiBox/VDE (Virtual Distribution Environment) models) has nearly limitless flexibility." ('861 2:37-40)</p> <p>15. "Such capabilities allow VDE supported product models to evolve by progressively reflecting requirements of 'next' participants in an electronic commercial models." ('193 297:12-15)</p>  |
| 25. | 193.1: "a budget specifying the number of copies which can be made of said digital file" | <p><b>Intrinsic:</b></p> <p>1. "For example, content control information for a given piece of content may be stipulated as senior information and therefore not changeable, might be put in place by a content creator and might stipulate that national distributors of a given piece of their content may be permitted to make 100,000 copies per calendar quarter, so long as such copies are provided to bonfire end-users, but may pass only a single copy of such content to a local retailers and the control information limits such a retailer to making no more than 1,000 copies per month for retail sales to end-users. In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer)." ('193 48:19-34)</p> <p>2. "... storing a first digital file and a first control in a first secure container, said first control constituting a first budget which governs the number of copies which may be made of said first digital file or a portion of said first digital file while said first digital file is contained in said first secure container," ('193 claim 60)</p> <p>3. "A certain content provider might, for example, require metering the number of copies made for distribution to employees of a given software program (a portion of the program might be maintained in encrypted form and require the presence of a VDE installation to run). This would require the execution of a metering method for copying of the property each time a copy was made for another employee." ('193 20:36-43)</p> <p>4. "For example, in the earlier example of a user with a desktop and a notebook computer, a provider may allow a user to make copies of information necessary to enable the notebook computer based on information present in the desktop</p> |

|     | Claim<br>Term/Phrase  | Evidence Supporting MS Construction  |
|-----|---|--|
|     |   | <p>computer, but not allow any further copies of said information to be made by the notebook VDE node. In this example, the distribution control structure described earlier would continue to exist on the desktop computer, but the copies of the enabling information passed to the notebook computer would lack the required distribution control structure to perform distribution from the notebook computer. Similarly, a distribution control structure may be provided by a content provider to a content provider who is a distributor in which a control structure would enable a certain number of copies to be made of a VDE content container object along with associated copies of permissions records, but the permissions records would be altered (as per specification of the content provider, for example) so as not to allow end-users who received distributor created copies from making further copies for distribution to other VDE nodes.” (‘193 264:29-49)</p> <p>5. “SPU 500 is enclosed within and protected by a ‘tamper resistant security barrier’ 502. Security barrier 502 separates the secure environment 503 from the rest of the world. It prevents information and processes within the secure environment 503 from being observed, interfered with and leaving except under appropriate secure conditions.” (‘193 59:48-53)</p> <p>6. “Secure container 302 may also contain an electronic, digital control structure 4078. This control structure 4078 (which could also be delivered independently in another container 302 different from the one carrying the image 4068I and/or the data 4068D) may contain important information controlling use of container 302. For example, controls 4078 may specify who can open container 302 and under what conditions the container can be opened. Controls 4078 might also specify who, if anyone, object 300 can be passed on to. As another example, controls 4078 might specify restrictions on how the image 4068I and/or data 4068D can be used (e.g., to allow the recipient to view but not change the image and/or data as one example). The detailed nature of control structure 4078 is described in connection, for example, with FIGS. 11D-11J ; FIG. 15 ; FIGS. 17-26B; and FIGS. 41A-61.” (‘683 25:62-26:10)</p> <p>7. “Many objects 300 that are distributed by physical media and/or by ‘out of channel’ means (e.g., redistributed after receipt by a customer to another customer) might not include key blocks 810 in the same object 300 that is used to transport the content protected by the key blocks. This is because VDE objects may contain data that can be electronically copied outside the confines of a VDE node. If the content is encrypted, the copies will also be encrypted and the copier cannot gain access to the content unless she has the appropriate decryption key(s).” (‘193 128:66)</p> <p>8. “Although block 1262 includes encrypted summary services information on the back up, it preferably does not include SPU device private keys, shared keys, SPU code and other internal security information to prevent this information from ever becoming available to users even in encrypted form.” (‘193 166:59-64)</p> |
| 26. | 193.1:<br>“controlling<br>the copies<br>made of said<br>digital file” | See above.   |

|     | Claim Term/Phrase   | Evidence Supporting MS Construction   |
|-----|---|---|
| 27. | 721.1:<br>“digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class” | <p><b>Intrinsic:</b></p> <ol style="list-style-type: none"> <li>1. “In one example, verifying authority 100 may digitally sign identical copies of load module 54 for use by different classes or ‘assurance levels’ of electronic appliances 61.” (‘721 18:19-22)</li> <li>2. “Protected execution spaces such as protected processing environments can be programmed or otherwise conditioned to accept only those load modules or other executables bearing a digital signature/certificate of an accredited (or particular) verifying authority. Tamper resistant barriers may be used to protect this programming or other conditioning. The assurance levels described below are a measure or assessment of the effectiveness with which this programming or other conditioning is protected.” (‘721 5:1-9)</li> <li>3. “For example, protected processing environments or other secure execution spaces that are more impervious to tampering (such as those providing a higher degree of physical security) may use an assurance level that isolates it from protected processing environments or other secure execution spaces that are relatively more susceptible to tampering (such as those constructed solely by software executing on a general purpose digital computer in a non-secure location).” (‘721 6:34-41)</li> <li>4. “The present invention may use a verifying authority and the digital signatures it provides to compartmentalize the different electronic appliances depending on their level of security (e.g., work factor or relative tamper resistance).” (‘721 6:53-56)</li> <li>5. “Assurance level I might be used for an electronic appliance(s) 61 whose protected processing environment 108 is based on software techniques that may be somewhat resistant to tampering. An example of an assurance level I electronic appliance 61A might be a general purpose personal computer that executes software to create protected processing environment 108. An assurance level II electronic appliance 61B may provide a protected processing environment 108 based on a hybrid of software security techniques and hardware-based security techniques. An example of an assurance level II electronic appliance 61B might be a general purpose personal computer equipped with a hardware integrated circuit secure processing unit (‘SPU’) that performs some secure processing outside of the SPU (see Ginter et al. patent disclosure FIG. 10 and associated text). Such a hybrid arrangement might be relatively more resistant to tampering than a software-only implementation. The assurance level III appliance 61C shown is a general purpose personal computer equipped with a hardware-based secure processing unit 132 providing and completely containing protected processing environment 108 (see Ginter et al. FIGS. 6 and 9 for example). A silicon-based special purpose integrated circuit security chip is relatively more tamper-resistant than implementations relying on software techniques for some or all of their tamper-resistance.” (‘721 6:44 – 7:5)</li> <li>6. “Assurance level in this example may be assigned to a particular protected processing environment 108 at initialization (e.g., at the factory in the case of hardware-based secure processing units). Assigning assurance level at initialization time facilitates the use of key management (e.g., secure key exchange protocols) to enforce isolation based on assurance level. For example,</li> </ol> |

|     | Claim Term/Phrase   | Evidence Supporting MS Construction  |
|-----|---|--|
|     |   | since establishment of assurance level is done at initialization time, rather than in the field in this example, the key exchange mechanism can be used to provide new keys (assuming an assurance level has been established correctly)." ('721 17:13-23)   |
| 28. | 891.1:<br>"securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item" | <b>Intrinsic:</b><br>1. "Such secure combination of VDE manage pieces of content will frequently require VDE's ability to securely derive content control information which accommodates the control information requirements, including any combinational rules, of the respective VDE managed pieces of content and reflects an acceptable agreement between plural control information sets." ('193 296:26-32)  |
| 29. | 900.155:<br>"derives information from one or more aspects of said host processing environment"  | <b>Intrinsic:</b><br>1. See '900 73:1- 80:6<br>a. "SPU Integrated Within CPU<br>b. As discussed above, it may be desirable to integrate CPU 654 and SPU 500 into the same integrated circuit and/or device. SPU 500 shown in FIG. 9 includes a microprocessor 520 that may be similar or identical to a standard microprocessor available off-the-shelf from a variety of manufacturers. Similarly, the SPU DMA controller 526 and certain other microprocessor support circuitry may be standard implementations available in off-the-shelf microprocessor and/or microcomputer chips. Since many of the general control and processing requirements provided by SPU 500 in the preferred embodiment can be satisfied using certain generic CPU and/or microcontroller components, it may be desirable to integrate SPU VDE functionality into a standard generic CPU or microcontroller chip. Such an integrated solution can result in a very cost-effective 'dual mode' component that is capable of performing all of the generic processing of a standard CPU as well as the secure processing of an SPU. Many of the control logic functions performed by the preferred embodiment SPU can be performed by generic CPU and/or micro-controller logic so that at least a portion of the control logic does not have to be duplicated. Additional cost savings (e.g., in terms of reducing manufacturing costs, inventory costs and printed circuit board real estate requirements) may also be obtained by not requiring an additional, separate physical SPU 500 device or package. FIG. 9A shows one example architecture of a combination CPU/SPU 2650. CPU/SPU |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>2650 may include a standard microprocessor or microcontroller 2652, a standard bus interface unit (BIU) 2656, and a standard (optional) DMA controller 2654, as well as various other standard I/O controllers, computation circuitry, etc. as may be found in a typical off-the-shelf microprocessor/microcontroller. Real time clock 528 may be added to the standard architecture to give the CPU/SPU 2650 access to the real time clock functions as discussed above in connection with FIG. 9. Real-time clock 528 must be protected from tampering in order to be secure. Such protections may include internal or external backup power, an indication that its power (and thus its operation) has been interrupted, and/or an indication that the external clock signal(s) from which it derives its timing have been interfered with (e.g., sped up, slowed down). Similarly, an encrypt/decrypt engine 522, pattern matching engine 524, compression/decompression engine 546 and/or arithmetic accelerator 544 may be added if desired to provide greater efficiencies, or the functions performed by these components could be provided instead by software executing on microprocessor 2652. An optional memory management unit 540 may also be provided if desired. A true random number generator 542 may be provided also if desired. Connections shown between mode interface switch 2658 and other components can carry both data and control information, specifically control information that determines what security-relevant aspects of the other components are available for access and/or manipulation.</p> <p>c. In addition, secure ROM 532 and/or secure RAM 534 may be provided within CPU/SPU 2650 along with a 'mode interface switch' 2658a, 2658b. Mode interface switch 2658 selectively provides microprocessor 2652 with access to secure memory 532, 534 and other secure components (blocks 522, 546, 524, 542, 544, 528) depending upon the 'mode' CPU/SPU 2650 is operating in. CPU/SPU 2650 in this example may operate in two different modes: an 'SPU' mode, or a 'normal' mode. In the 'normal' mode, CPU/SPU 2650 operates substantially identically to a standard off-the-shelf CPU while also protecting the security of the content, state, and operations of security-relevant components included in CPU/SPU 2650. Such security-relevant components may include the secure memories 532, 534; the encrypt/decrypt engine 522, the optional pattern-matching engine 524, random number generator 542, arithmetic accelerator 544, the SPU-not-initialized flag 2671, the secure mode interface switch 2658, the real-time clock 528, the DMA controller 2654, the MMU 540, compress/decompress block 546, and/or any other components that may affect security of the operation of the CPU/SPU in 'SPU' mode.</p> <p>d. In this example, CPU/SPU 2650 operating in the 'normal' mode controls mode interface switch 2658 to effectively 'disconnect' (i.e., block unsecure access to) the security-relevant components, or to the security-relevant aspects of the operations of such components as have a function for both 'normal' and 'SPU' mode. In the 'normal' mode, for</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>example, microprocessor 2652 could access information from standard registers or other internal RAM and/or ROM (not shown), execute instructions in a 'normal' way, and perform any other tasks as are provided within a standard CPU—but could not access or compromise the contents of secure memory 532, 534 or access blocks 522, 524, 542, 544, 546. In this example 'normal' mode, mode interface switch 2658 would effectively prevent any access (e.g., both read and write access) to secure memory 532, 534 so as to prevent the information stored within that secure memory from being compromised.</p> <p>e. When CPU/SPU 2650 operates in the 'SPU' mode, mode interface switch 2658 allows microprocessor 2652 to access secure memory 532, 534, and to control security-relevant aspects of other components in the CPU/SPU. The 'SPU' mode in this example requires all instructions executed by microprocessor 2652 to be fetched from secure memory 532, 534—preventing execution based on 'mixed' secure and non-secure instructions. In the 'SPU' mode, mode interface switch 2658 may, in one example embodiment, disconnect or otherwise block external accesses carried over bus 652 from outside CPU/SPU 2650 (e.g., DMA accesses, cache coherency control accesses) to ensure that the microprocessor 2652 is controlled entirely by instructions carried within or derived from the secure memory 532, 534. Mode interface switch 2658 may also disconnect or otherwise block access by microprocessor 2652 to some external memory and/or other functions carried over bus 652. Mode interface switch 2658 in this example prevents other CPU operations/instructions from exposing the contents of secure memory 532, 534.</p> <p>f. In the example shown in FIG. 9A, the mode control of mode interface switch 2658 is based on a 'mode' control signal provided by microprocessor 2652. In this example, microprocessor 2652 may be slightly modified so it can execute two 'new' instructions: 'enable 'SPU' mode' instruction, and 'disable 'SPU' mode' instruction.</p> <p>g. When microprocessor 2652 executes the 'enable 'SPU' mode' instruction, it sends an appropriate 'mode' control signal to mode interface switch 2658 to 'switch' the interface switch into the 'SPU' mode of operation. When microprocessor 2652 executes the 'disable 'SPU' mode' instruction, it sends an appropriate 'mode' control signal to mode interface switch 2658 to disable the 'SPU' mode of operation.</p> <p>h. When CPU/SPU 2650 begins operating in the 'SPU' mode (based on microprocessor 2652 executing the 'enable 'SPU' mode' instruction), mode interface switch 2658 forces microprocessor 2652 to begin fetching instructions from secure memory 532, 534 (e.g., beginning at some fixed address) in one example. When CPU/SPU 2650 begins operating in this example 'SPU' mode, mode interface switch 2658 may force microprocessor 2652 to load its registers from some fixed address in secure memory 532, 534 and may begin execution based on such register content. Once operating in the 'SPU' mode, microprocessor 2652 may</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>provide encryption/decryption and other control capabilities based upon the code and other content of secure memory 532, 534 needed to provide the VDE functionality of SPU 500 described above. For example, microprocessor 2652 operating under control of information within secure memory 532, 534 may read encrypted information from bus 652 via bus interface unit 2656, write decrypted information to the bus interface unit, and meter and limit decryption of such information based on values stored in the secure memory.</p> <p>i. At the end of secure processing, execution by microprocessor 2652 of the 'disable SPU mode' instruction may cause the contents of all registers and other temporary storage locations used by microprocessor 2652 that are not within secure memory 532, 534 to be destroyed or copied into secure memory 532, 534 before 'opening' mode interface switch 2658. Once mode interface switch 2658 is 'open,' the microprocessor 2652 no longer has access to secure memory 532, 534 or the information it contained, or to control or modify the state of any other security-relevant components or functions contained within CPU/SPU 2650 to which access is controlled by mode interface switch 2658.</p> <p>j. Whenever CPU/SPU 2650 enters or leaves the 'SPU' mode, the transition is performed in such a way that no information contained in the secure memory 532, 534 or derived from it (e.g., stored in registers or a cache memory associated with microprocessor 2652) while in the 'SPU' mode can be exposed by microprocessor 2652 operations that occur in the 'normal' mode. This may be accomplished either by hardware mechanisms that protect against such exposure, software instructions executed in 'SPU' mode that clear, reinitialize, and otherwise reset during such transitions, or a combination of both.</p> <p>k. In some example implementations, interrupts may be enabled while CPU/SPU 2650 is operating in the 'SPU' mode similarly interrupts and returns from interrupts while in the 'SPU' mode may allow transitions from 'SPU' mode to 'normal' mode and back to 'SPU' mode without exposing the content of secure memory 532, 534 or the content of registers or other memory associated with microprocessor 2652 that may contain information derived from secure mode operation.</p> <p>l. In some example implementations, there may be CPU/SPU activities such as DMA transfers between external memory and/or devices and secure memory 532, 534 that are initiated by microprocessor 2652 but involve autonomous activity by DMA controller 2654 and, optionally, encrypt/decrypt engine 522 and/or compress/decompress engine 546. In such implementations, mode interface switch 2658 and its associated control signals may be configured to permit such pending activities (e.g. DMA transfers) to continue to completion even after CPU/SPU 2650 leaves 'SPU' mode, provided that upon completion, all required clearing, reinitialization, and/or reset activities occur, and provided that no access or interference is permitted with the pending activities except when CPU/SPU 2650 is operating in 'SPU' mode.</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>m. In an additional example embodiment, encryption/decryption logic may be connected between microprocessor 2652 and secure memory 532, 534. This additional encryption/decryption logic may be connected 'in parallel' to mode interface switch 2658. The additional encryption/decryption logic may allow certain accesses by microprocessor 2652 to the secure memory 532, 534 when CPU/SPU 2650 is operating in the 'normal' mode. In this alternate embodiment, reads from secure memory 532, 534 when CPU/SPU 2650 is operating in the 'normal' mode automatically result in the read information being encrypted before it is delivered to microprocessor 2652 (and similarly, and writes to the secure memory may result in the written information being decrypted before it is deposited into the secure memory). This alternative embodiment may permit access to secure memory 532, 534 (which may in this example store the information in 'clear' form) by microprocessor 2652 when CPU/SPU 2650 is operating in the 'non-secure normal' mode, but only reveals the secure memory contents to microprocessor 2652 in unencrypted form when the CPU/SPU is operating in the 'SPU' mode. Such access may also be protected by cryptographic authentication techniques (e.g., message authentication codes) to prevent modification or replay attacks that modify encrypted data stored in secure memory 532, 534. Such protection may be performed utilizing either or both of software and/or hardware cryptographic techniques.</p> <p>n. All of the components shown in FIG. 9A may be disposed within a single integrated circuit package. Alternatively, mode interface switch 2658 and secure memory 532, 534, and other security-relevant components might be placed within an integrated circuit chip package and/or other package separate from the rest of CPU/SPU 2650. In this two-package version, a private bus could be used to connect microprocessor 2652 to the mode interface switch 2658 and associated secure memory 532, 534. To maintain security in such multi-package versions, it may be necessary to enclose all the packages and their interconnections in an external physical tamper-resistant barrier.</p> <p>o. Initialization of Integrated CPU/SPU</p> <p>p. Instructions and/or data may need to be loaded into CPU/SPU 2650 before it can operate effectively as an SPU 500. This may occur during the manufacture of CPU/SPU 2650 or subsequently at a CPU/SPU initialization facility. Security of such initialization may depend on physical control of access to the CPU/SPU component(s), on cryptographic means, or on some combination of both. Secure initialization may be performed in plural steps under the control of different parties, such that an initialization step to be performed by party B is preconditioned on successful performance of a step by party A. Different initialization steps may be protected using different security techniques (e.g. physical access, cryptography).</p> <p>q. In this example, switch 2658 may expose an external control signal 2670 that requests operation in 'SPU' mode rather than 'normal'</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>mode after a power-on reset. This signal would be combined (e.g., by a logical AND 2672) with a non-volatile storage element 2671 internal to CPU/SPU 2650. If both of these signals are asserted, AND gate 2672 would cause CPU/SPU 2650 to begin operating in SPU mode, either executing existing instructions from an address in SPU memory 532, executing instructions from main memory 2665 or otherwise external to the CPU/SPU. The instructions thus executed would permit arbitrary initialization and other functions to be performed in 'SPU' mode without necessarily requiring any instructions to be previously resident in the SPU memory 532.</p> <p>r. Once initialized, the SPU would, under control of its initialization program, indicate to switch 2658 that the flag 2671 is to be cleared. Clearing flag 2671 would permanently disable this initialization capability because no mechanism would be provided to set flag 2671 back to its initial value. If flag 2671 is clear, or control signal 2670 is not asserted, CPU/SPU 2650 would behave precisely as does microprocessor 2652 with respect to power-on reset and other external conditions. Under such conditions, only execution of the 'enable SPU mode' instruction or otherwise requesting SPU mode under program control would cause 'SPU' mode to be entered.</p> <p>s. Additionally, a mechanism could be provided to permit microprocessor 2652 and/or control signal 2672 to reinitialize the flag 2671. Such reinitialization would be performed in a manner that cleared secure memory 532, 534 of any security-relevant information and reinitialized the state of all security-relevant components. This reinitialization mechanism would permit CPU/SPU 2650 to be initialized several times, facilitating testing and/or re-use for different applications, while protecting all security-relevant aspects of its operation.</p> <p>t. In the preferred embodiment, CPU/SPU 2650 would, when SPU mode has not yet been established, begin operating in SPU mode by fetching instructions from secure non-volatile memory 532, thereby ensuring a consistent initialization sequence and preventing SPU dependence on any information held outside CPU/SPU 2650. This approach permits secret initialization information (e.g., keys for validating digital signatures on additional information to be loaded into secure memory 532, 534) to be held internally to CPU/SPU 2650 so that it is never exposed to outside access. Such information could even be supplied by a hardware 'mask' used in the semiconductor fabrication process.</p> <p>u. CPU/SPU Integrated With Unmodified Microprocessor</p> <p>v. FIG. 9B shows an additional example embodiment, in which a completely standard microprocessor 2652 integrated circuit chip could be transformed into a CPU/SPU 2650 by adding an SPU chip 2660 that mediates access to external I/O devices and memory. In such an embodiment, the microprocessor 2652 would be connected to the SPU chip 2660 by a private memory bus 2661, and all three such components would be contained within hardware tamper-resistant barrier 502.</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>w. In this embodiment, SPU chip 2660 may have the same secure components as in FIG. 9, i.e., it may have a ROM/EEPROM 532, a RAM 532, an RTC 528, an (optional) encryption/decryption engine 522, an (optional) random number generator (RNG) 542, an (optional) arithmetic accelerator 544, and a (optional) compression/decompression engine 546, and a (optional) pattern matching circuit 524. Microprocessor 520 is omitted from SPU chip 2660 since the standard microprocessor 2650 performs the processing functions instead. In addition, SPU chip 2660 may include a flag 2671 and AND gate logic 2672 for the initialization purposes discussed above.</p> <p>x. In addition, SPU chip 2660 includes an enhanced switch 2663 that provides the same overall (bus enhanced) functionality performed by the switch 2658 in the FIG. 9A embodiment.</p> <p>y. Enhanced switch 2663 would perform the functions of a bus repeater, mediator and interpreter. For example, enhanced switch 2663 may act as a bus repeater that enables microprocessor 2652's memory accesses made over internal memory bus 2661 to be reflected to external memory bus 2664 and performed on main memory 2665. Enhanced switch 2663 may also act as a bus repeater similarly for internal I/O bus 2662 to external I/O bus 2665 in the event that microprocessor 2652 performs I/O operations distinctly from memory operations. Enhanced switch 2663 may also perform the function of a mediator for microprocessor control functions 2666 (e.g., non-maskable interrupt, reset) with respect to externally requested control functions 2667. Enhanced switch 2663 may also provide mediation for access to SPU-protected resources such as ROM 532, RAM 534, encrypt/decrypt engine 522 (if present), random number generator 542 (if present), arithmetic accelerator 544 (if present), pattern matching engine 524 (if present), and real-time clock 528 (if present). Enhanced switch 2663 may also act as an interpreter of control signals received from microprocessor 2652 indicating entry to, exit from, and control of SPU mode.</p> <p>z. Switch 2663 in this example recognizes a specific indication (e.g., an instruction fetch access to a designated address in the secure memory 532) as the equivalent to the 'enable 'SPU' mode' instruction. Upon recognizing such an indication, it may isolate the CPU/SPU 2650 from external buses and interfaces 2664, 2665, and 2667 such that any external activity, such as DMA cycles, would be 'held' until the switch 2663 permits access again. After this, switch 2663 permits a single access to a specific location in secure memory 532 to complete.</p> <p>aa. The single instruction fetched from the designated location performs a control operation (a cache flush, for example), that can only be performed in microprocessor 2652's most privileged operating mode, and that has an effect visible to switch 2663. Switch 2663 awaits the occurrence of this event, and if it does not occur within the expected number of cycles, does not enter 'SPU' mode.</p> <p>bb. Occurrence of the control operation demonstrates that</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>microprocessor 2652 is executing in its most privileged 'normal' mode and therefore can be trusted to execute successfully the 'enter 'SPU' mode' sequence of instructions stored in secure memory 532. If microprocessor 2652 were not executing in its most privileged mode, there would be no assurance that those instructions would execute successfully. Because switch 2663 isolates microprocessor 2652 from external signals (e.g., interrupts) until 'SPU' mode is successfully initialized, the entry instructions can be guaranteed to complete successfully.</p> <p>cc. Following the initial instruction, switch 2663 can enter 'partial SPU mode,' in which a restricted area of ROM 532 and RAM 534 may be accessible. Subsequent instructions in secure memory 532 may then be executed by microprocessor 2652 to place it into a known state such that it can perform SPU functions—saving any previous state in the restricted area of RAM 534 that is accessible. After the known state is established, an instruction may be executed to deliver a further indication (e.g., a reference to another designated memory location) to switch 2663, which would enter 'SPU' mode. If this further indication is not received within the expected interval, switch 2663 will not enter 'SPU' mode. Once in 'SPU' mode, switch 2663 permits access to all of ROM 532, RAM 534, and other devices in SPU chip 2660.</p> <p>dd. The instructions executed during 'partial SPU' mode must be carefully selected to ensure that no similar combination of instructions and processor state could result in a control transfer out of the protected SPU code in ROM 532 or RAM 534. For example, internal debugging features of microprocessor 2652 must be disabled to ensure that a malicious program could not set up a breakpoint later within protected SPU code and receive control. Similarly, all address translation must be disabled or reinitialized to ensure that previously created MMU data structures would not permit SPU memory accesses to be compromised. The requirement that the instructions for 'partial SPU mode' run in the microprocessor 2652's most privileged mode is necessary to ensure that all its processor control functions can be effectively disabled.</p> <p>ee. The switch 2663 provides additional protection against tampering by ensuring that the expected control signals occur after an appropriate number of clock cycles. Because the 'partial SPU' initialization sequence is entirely deterministic, it is not feasible for malicious software to interfere with it and still retain the same timing characteristics, even if malicious software is running in microprocessor 2652's most privileged mode.</p> <p>ff. Once in 'SPU' mode, switch 2663 may respond to additional indications or signals generated by microprocessor 2652 (e.g., references to specific memory addresses) controlling features of SPU mode. These might include enabling access to external buses 2664 and 2665 so that SPU-protected code could reference external memory or devices. Any attempts by components outside CPU/SPU 2650 to perform operations</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>(e.g., accesses to memory, interrupts, or other control functions) may be prevented by switch 2663 unless they had been explicitly enabled by instructions executed after 'SPU' mode is entered. To leave SPU mode and return to normal operation, the instructions executing in 'SPU' mode may provide a specific indication to switch 2663 (e.g., a transfer to a designated memory address). This indication may be recognized by switch 2663 as indicating a return to 'normal mode,' and it may again restrict access to ROM 532, RAM 534, and all other devices within SPU chip 2660, while re-enabling external buses and control lines 2664, 2665, and 2667. The instructions executed subsequently may restore the CPU state to that which was saved on entry to SPU mode, so that microprocessor 2652 may continue to perform functions in progress when the SPU was invoked.</p> <p>gg. In an alternate embodiment, the entry into SPU mode may be conditioned on an indication recognized by switch 2663, but the switch may then use a hardware mechanism (e.g., the processor's RESET signal) to reinitialize microprocessor 2562. In such an embodiment, switch 2663 may not implement partial SPU mode, but may instead enter SPU mode directly and ensure that the address from which instructions would be fetched by microprocessor 2652 (specific to microprocessor 2652's architecture) results in accesses to appropriate locations in the SPU memory 532. This could reduce the complexity of the SPU mode entry mechanisms in switch 2663, but could incur an additional processing cost from using a different reinitialization mechanism for microprocessor 2652.</p> <p>hh. SPU chip 2660 may be customized to operate in conjunction with a particular commercial microprocessor. In this example, the SPU may be customized to contain at least the specialized 'enter SPU mode' instruction sequences to reinitialize the processor's state and, to recognize special indications for SPU control operations. SPU chip 2660 may also be made electrically compatible with microprocessor 2652's external bus interfaces. This compatibility would permit CPU/SPU 2650 to be substituted for microprocessor 2652 without change either to software or hardware elsewhere in a computer system.</p> <p>ii. In other alternate embodiments, the functions described above for SPU chip 2600, microprocessor 2652, and internal buses 2661, 2662, and 2666 could all be combined within a single integrated circuit package, and/or on a single silicon die. This could reduce packaging complexity and/or simplify establishment of the hardware tamper-resistant barrier 502.</p> <p>jj. The hardware configuration of an example of electronic appliance 600 has been described above. The following section describes an example of the software architecture of electronic appliance 600 provided by the preferred embodiment, including the structure and operation of preferred embodiment 'Rights Operating System' ('ROS') 602."</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>2. See '900 230:55 – 233:34</p> <ul style="list-style-type: none"> <li>a. "Integrity of Software-Based PPE Security</li> <li>b. As discussed above in connection with FIG. 10, some applications may use a software-based protected processing environment 650 (such as a 'host event processing environment' (HPE) 655) providing a software-based tamper resistant barrier 674. Software-based tamper resistant barrier 674 may be created by software executing on a general-purpose CPU. Various software protection techniques may be used to construct and/or provide software-based tamper resistant barrier 674.</li> <li>c. The risks or threat of attacks described above in connection with PPE 650 apply to a software-based PPE. An important threat to be countered with respect to a software-based tamper resistant barrier 674 is an attack based on a distributable computer program that can defeat the tamper resistant barrier wherever the program is run. Since a software-based tamper resistant barrier 674 typically will not be as secure as a hardware-based tamper resistant barrier 502, it is useful to explore example steps and procedures a 'cracker' might use to "crack" a software'-based tamper resistant barrier.</li> <li>d. FIGS. 67A and 67B show example 'cracking' techniques a 'cracker' might use to attack software-based tamper resistant barrier 674.</li> <li>e. Referring to FIG. 67A, the software used to create tamper resistant barrier 674 may be distributed, for example, on a storage medium 3370 such as a floppy diskette or optical disk (or, this software could be distributed electronically over network 108 and stored locally in a computer memory). The software distribution medium 3370 provides software (code and data) for loading into a computing device such as a general purpose personal computer 3372, for example. Personal computer 3372 may include, for example, a random access memory 3374 and a hard disk 3376.</li> <li>f. In one example, the software distribution medium 3370 might include installation materials 3470 and operational materials 3472. The installation materials 3470 may be executed by computer 3372 to install the operational materials 3472 onto the computer's hard disk 3376. The computer 3372 may then execute the operational materials 3472 from its hard disk 3376 to provide software-based protected processing environment 650 and associated software-based tamper resistant barrier 672.</li> <li>g. In this example, one attack technique an attacker might use is to analyze software distribution medium 3370 (see FIG. 67B, block 3352). Such analysis can take many forms.</li> <li>h. Such analysis could be performed by a combination of one or more techniques. Such techniques include, but are not limited to, the following:</li> <li>i. An attacker can manually 'dump' and/or disassemble listings of the data from medium 3370. This analysis is represented in FIG. 67A by magnifying glass 3352A.</li> <li>j. An attacker can use cryptanalytic and/or key search techniques to</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>decrypt any encrypted data from medium 3370.</p> <ul style="list-style-type: none"> <li>k. An attacker can use automated or semi-automated disassembly tools to explore the functions of programs stored on medium 3370 by studying the operation and flow of the assembly language representation of the programs. This analysis is represented in FIG. 67A by block 3352B.</li> <li>l. An attacker can use software reverse-engineering tools to reconstruct high-level language representations of the programs on medium 3370, and study their functions. This analysis is represented in FIG. 67A by block 3352C, producing source code 3371.</li> <li>m. An attacker can use software reverse-engineering tools to create an equivalent program to the programs stored on medium 3370. As the equivalent program may be in a more convenient form, possibly in a higher-level language, it may be more amenable to analysis. This analysis is also represented in FIG. 67A by block 3352C, producing source code 3371.</li> <li>n. An attacker can use software debugging and/or simulation tools to follow and/or modify the dynamic execution of programs from medium 3370. This technique can be combined with any of the above static analysis techniques to study the program as it operates. This analysis is represented in FIG. 67A by block 3352B.</li> <li>o. An attacker can use hardware-based debugging and/or simulation tools (e.g., an in-circuit emulator, or ICE) to follow and/or modify the dynamic execution of programs from medium 3370. This technique may be more effective than the equivalent using software debugging and/or simulation tools because it has less potential effect on operation of the programs. This analysis is represented in FIG. 67A by block 3352B.</li> <li>p. Such analysis could provide clues and insights into the installation materials 3470, the operational materials 3472, or both.</li> <li>q. Another attack technique could focus on the operational materials 3472 in the form in which they are installed on personal computer 3372. For example, one form of analysis might involve analyzing the on-disk copy of the installed software and/or associated data files installed on computer hard disk 3376 (see FIG. 67B, block 3354). This analysis is represented in FIG. 67A as a magnifying glass 3354B. Because the installed operational materials 3472 can be executed by computer 3372, the analysis need not be limited to analyzing the static information stored on hard disk 3376, but could involve performing static and/or dynamic analysis of the executing software (see FIG. 67B, blocks 3356, 3358). Any of the techniques described above could be used to analyze the operational material software 3472 to yield source code or other more interpretable form 3373A and/or a memory image 3373B. The static and/or dynamic data within RAM 3374A could be similarly analyzed (see FIG. 67A, magnifying glass 3354A).</li> <li>r. The resulting source code 3373A and/or memory image 3373B could be carefully analyzed and reviewed (see magnifying glasses 3354D, 3354E) to obtain an understanding of both the static and dynamic structure and</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>operation of operational materials 3272. Dynamic code analysis could involve, for example, tracing, single-stepping, data, or code break points of the executing software image, using analysis techniques such as described above. The executing software could be modified dynamically (for example, by patching) during normal operation to attempt to bypass its protection mechanisms and/or to learn more about how it operates (see FIG. 67B, block 3360, and the 'changes' inserted into FIG. 67A memory image 3373B).</p> <ul style="list-style-type: none"> <li>s. A further attack technique in this example might involve comparing installed operational material 3472 software and data files among several different PPE 650 instances to identify important data structures, such as cryptographic keys (see 'compare' block 3362A of FIG. 67A; and FIG. 67B, block 3362). The resulting list of differences 3362B could be carefully analyzed (see FIG. 67A's magnifying glass 3362C) to obtain important clues, using analysis techniques such as described above.</li> <li>t. A further attack technique might involve comparing the memory and/or disk images of installed operational material 3472 software and data files in a single instance of PPE 650, after performing various operations using the PPE. This could serve to identify important data structures, such as cryptographic keys (see 'compare' block 3362A of FIG. 67A; and FIG. 67B, block 3362). The resulting list of differences 3362B could be carefully analyzed (see FIG. 67A's magnifying glass 3362C) to obtain important clues, using analysis techniques such as described above.</li> <li>u. A further attack technique might involve analyzing the timing and/or order of modification to memory and/or disk images of installed operational material 3472 software and data files in a single instance of PPE 650, during the performance performing various operations using the PPE. This could serve to identify important data structures, such as cryptographic keys (see 'compare' block 3362A of FIG. 67A; and FIG. 67B, block 3362). The resulting list of differences 3362B could be carefully analyzed (see FIG. 67A's magnifying glass 3362C) to obtain important clues, using analysis techniques such as described above.</li> <li>v. A further attack technique might involve duplicating one installed operational material 3472 instance by copying the programs and data from one personal computer 3372B to another personal computer 3372C or emulator (see FIG. 67B, block 3364, and the 'copy' arrow 3364A in FIG. 67A). The duplicated PPE instance could be used in a variety of ways, such as, for example, to place an impostor PPE 650 instance on-line and/or to permit further dynamic analysis.</li> <li>w. A still additional avenue of attack might involve, for example, saving the state of a PPE 650 (see FIG. 67A, block 3366B)--for example, before the expenditure of credit--and restoring the state at a subsequent time (e.g., after a payment operation occurs) (see FIG. 67A, arrows 3366A, 3366C, and FIG. 67B, block 3366). The stored state information 3366B may also be analyzed (see FIG. 67A, magnifying glass 3354F).</li> <li>x. No software-only tamper resistant barrier 674 can be wholly effective</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>against all of these threats. A sufficiently powerful dynamic analysis (such as one employing an in-circuit emulator) can lay bare all of the software-based PPE 650's secrets. Nonetheless, various techniques described below in connection with FIG. 69A and following make such an analysis extremely frustrating and time consuming--increasing the 'work factor' to a point where it may become commercially unfeasible to attempt to 'crack' a software-based tamper resistant barrier 674."</p> <p>3. See '900 235:28 – 244:15</p> <ul style="list-style-type: none"> <li>a. "Example Techniques for Forming Software-Based Tamper Resistant Barrier</li> <li>b. Various software protection techniques detailed above in connection with FIG. 10 may provide software-based tamper resistant barrier 674 within a software-only and/or hybrid software/hardware protected processing environment 650. The following is an elaboration on those above-described techniques. These software protection techniques may provide, for example, the following:</li> <li>c. An on-line registration process that results in the creation of a shared secret between the registry and the PPE 650 instance--used by the registry to create content and transactions that are meaningful only to that specific PPE instance.</li> <li>d. An installation program (that may be distinct from the PPE operational material software) that creates a customized installation of the PPE software unique to each PPE instance and/or associated electronic appliance 600.</li> <li>e. Camouflage protections that make it difficult to reverse engineer the PPE 650 operational materials during PPE operation.</li> <li>f. Integrity checks performed during PPE 650 operation (e.g., during on-line interactions with trusted servers) to detect compromise and minimize damage associated with any compromise.</li> <li>g. In general, the software-based tamper resistant barrier 674 may establish 'trust' primarily through uniqueness and complexity. In particular, uniqueness and customization complicate the ability of an attacker to: make multiple PPE instances with the same apparent identity; make it harder for an attacker to create a software program(s) that will defeat the tamper-resistant barrier 674 of multiple PPE instances; make it harder for the attacker to reverse engineer (e.g., based upon encryption so that normal debugging/emulation and other software testing tools can't easily provide access); and make it more difficult for an attacker to compare multiple PPE instances to determine differences between them.</li> <li>h. In addition, the overall software-based tamper resistant barrier 674 and associated PPE system is sufficiently complex so that it is difficult to tamper with a part of it without destroying other aspects of its functionality (i.e., a 'defense in depth'). Camouflaging techniques complicate an attacker's analysis through use of debugging/emulation or other software tools. For example, the PPE 650 may rewrite or overwrite</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>memory locations immediately after using same to make their contents unavailable for scrutiny. Similarly, the PPE 650 operational software may use hardware and/or time dependent sequences to prevent emulation. Additionally, some of the PPE 650 environment code may be self-modifying. These and other techniques make it much harder to crack an individual PPE 650 instance, and more importantly--much harder to write a program that could be used to defeat security on multiple PPE instances. Because the legitimate owner/user of a particular PPE instance may be trying to attack the security of his own system, these techniques assume that individual instances may eventually be cracked and provide additional security and safeguards that prevent (or make it more difficult) for the attacker who has cracked one PPE instance to use that information successfully in cracking other PPE instances. Specifically, these security techniques make it unlikely that an attacker who has successfully cracked one or a small number of PPE instances can write a program capable of compromising the security of any arbitrary other PPE instance, for example.</p> <ul style="list-style-type: none"> <li>i. Example Installation Process</li> <li>j. Briefly, the preferred example software-based PPE 650 installation process provides the following security techniques:<br/> encrypted software distribution,<br/> installation customized on a unique instance and/or electronic appliance basis,<br/> encrypted on-disk form,<br/> installation tied to payment method,<br/> unique software and data layout, and<br/> identifiable copies.</li> <li>k. FIG. 69A shows one example technique for distributing the PPE 650 software. In this example, the PPE 650 software is distributed as two separate parts and/or media: the installation materials 3470, and the operational materials 3472. Installation materials 3470 may provide executable code and associated data structures for installing the operational materials 3472 onto a personal computer hard disk 3376, for example (see FIG. 67A). The operational materials 3472 may provide executable code and associated data structures for providing protected processing environment 650 and associated software-based tamper resistant barrier 674.</li> <li>l. In this example, installation materials 3470 and operational materials 3472 are each encrypted by a 'deliverable preparation' process 3474 to provide encrypted installation materials 3470E and encrypted operational materials 3472E (the encrypted portions are indicated in FIG. 69A, by cross-hatching). In this example, a small portion 3470C of the installation materials 3470 may be maintained in clear (unencrypted) form to provide an initial portion of the installation routine that may be executed without decryption. This plain text portion 3470C may, for example, provide an initial dialog, using an encrypted or other secure protocol with a trusted</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>registry 3476 such as VDE administrator 200h for example. This makes the distributed installation materials 3470 and operational materials 3472 meaningless and unreadable to an attacker without additional information since the entire content (except for the initial dialog with the registry 3476) is unreadable.</p> <p>m. In this example, the 'deliverable preparation' process 3474 may encrypt the installation materials 3470 and operational materials 3472 using one or more secret keys known to the registry 3476. Multiple versions of these installation materials 3470 and operational materials 3472 may be distributed using different, secret keys so that compromise of one key exposes only a subset of the software distribution to unwanted disclosure. The only non-encrypted part of the software distribution in plaintext is that portion 3470C of installation materials 3470 used to establish initial contact with the registry 3476.</p> <p>n. The registry 3476 maintains a copy of the corresponding decryption keys within a key generation and cataloging structure 3478. It provides these keys on demand during the registration process (e.g., using a secure key exchange protocol, for example) to only legitimate users authorized to set up a new protected processing environment 650.</p> <p>o. FIGS. 69B-69C show example steps that may be performed by a installation routine 3470 to install a protected processing environment 650. In this example, upon coupling the installation materials 3470 to an electronic appliance 600 such as a personal computer 3372, the appliance begins executing the unencrypted installation materials portion 3470C. This plain text portion 3470C controls appliance 600 to contact registry 3476 and establish a registry dialog (FIG. 69B, block 3470(1)). The appliance 600 and the registry 3476 use a secure key exchange protocol to exchange installation keys so that the registry may deliver the appropriate installation key to the appliance (FIG. 69B, block 3470(2)). Using the provided installation key(s), the appliance 600 may decrypt and run additional portions of encrypted installation materials 3470E (FIG. 69B, block 3470(3) and following). Based on this additional installation program execution, appliance 600 may decrypt and install encrypted operational materials 3472E (FIG. 69B, block 3470(4)).</p> <p>p. Rather than simply installing the operational materials 3472, in one example, installation materials 3470 makes the installation different for each PPE 650 instance. For example, the installation materials 3470 may customize the installation by:</p> <ul style="list-style-type: none"> <li>uniquely embedding important data into the installed software,</li> <li>uniquely encrypting the installed software,</li> <li>uniquely making random changes to the installed software,</li> <li>uniquely mating the installed software with a particular electronic appliance 600,</li> <li>providing a unique static and/or dynamic layout or other structure.</li> </ul> <p>q. Randomly Embedded Cryptographic Keys</p> <p>r. Installation routine 3470 may, for example, modify the operational</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>materials 3472 to customize embedded locations where critical data such as cryptographic keys are stored. These keys may be embedded into the text of the operational materials 3472 at locations that vary with each installation. In this example, the registry 3476 may choose, on a random or pseudo-random basis, at least some of the operational material 3472 locations in which a particular installation routine 3470 may embed cryptographic keys or other critical data (see FIG. 69B, block 3470(5)).</p> <ul style="list-style-type: none"> <li>s. The installation process for the operational software may involve decrypting its distribution (which may be the same for all end users) and modifying it to encode the specific locations where its critical data (e.g., cryptographic keys) are stored. These keys may be embedded within the text of the program at locations that vary with every installation. The distribution of unique information into the operational software 3472 can be based on a secret key known to the registry 3476. This key may be communicated by the registry 3476 during the registration dialog using a secure key exchange. The key is shared between the registry 3476 and the PPE 650 instance, and can serve both to organize the installed PPE software, and as the basis of subsequent integrity checks.</li> <li>t. As shown in FIG. 69D, the operational materials 3472 may include embedded locations 3480(a), 3480(b), 3480(c), 3480(d), 3480(e), ... reserved for storing (embedding) critical information such as cryptographic keys. Each of these locations 3480 may initially store a random number string. In one example, the registry 3476 or installation routine 3470 performs a random operation 3482 to randomly select which subset of these locations 3480 is to be used by a particular instance for storing critical data. This selection list 3484 is applied as an input to an operation materials preparation step 3474a (part of the deliverable preparation operation 3474 shown in FIG. 69A). The operation materials preparation step 3474a also accepts, as an input, cryptographic keys from a secure key store 3486. In this example, the operation materials preparation step 3474a embeds the cryptographic keys provided by key store 3486 into the selected locations 3484 of operation materials 3472.</li> <li>u. In accordance with one example, the random operation 3482 selects a subset that is much less than all of the possible locations 3480--and the locations 3480 not used for storing cryptographic keys store random data instead. An attacker attempting to analyze installed operational materials 3472 won't be able to tell the difference between real cryptographic keys and random number strings inserted into a place where cryptographic keys might be stored.</li> <li>v. In this example, the random location selection 3484 (which is unique for each installation) may itself be encrypted by block 3488 based on an installation-unique key provided by key generation block 3490 for example. The encryption key may be securely maintained at registry 3476 so that the registry may later notify the installation materials 3470 of this key--allowing the installation materials to decrypt the resulting encrypted key location block 3492 and recover listing 3484 of the subset</li> </ul> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>of locations 3480 used for embedding cryptographic keys.</p> <p>w. Embedded Customized Random Changes</p> <p>x. Referring once again to FIG. 69B, the installed operational materials 3472 may be further customized for each instance by making random changes to reserved, unused portions of the operational materials (FIG. 69B, block 3470(6)). An example of this is shown in FIG. 69E. In this example, the operational materials 3472 include unused, embedded random data or code portions 3494. Another technique with similar effect is shown in FIG. 69F. In this example, false code sections 3496 are included within reserved areas of the operational materials 3472. These false code sections 3496 add complexity, and may also be used as a electronic 'fingerprint' to help trace copies. Because the false code sections 3496 are executable program code that are never executed (or if executed perform no actual functions other than confounding analysis by, for example, creating, modifying and/or destroying data that has no impact on the operation of PPE 650 but may appear to have such an impact), they can be used to confound analysis because they may be difficult for an attacker to distinguish from true code sections. In addition other false code may have the effect of disabling the execution of PPE 650 if executed. Correspondence Between Installed Software and Appliance 'Signature'. Another technique that may be used during the installation routine 3470 is to customize the operational materials 3472 by embedding a 'machine signature' into the operational materials to establish a correspondence between the installed software on a particular electronic appliance 600 (FIG. 69C, block 3470(7)). This technique prevents a software-based PPE 650 from being transferred from one electronic appliance 600 to another (except through the use of the appropriate secure, verified backup mechanism).</p> <p>y. For electronic appliances 600 where it is feasible to do so, the installation procedure 3470 may determine unique information about the electronic appliance 600 (e.g., a 'signature' SIG in the sense of a unique value—not necessarily a 'digital signature' in the cryptographic sense). Installation routine 3470 embeds the electronic appliance 'signature' SIG in the installed operational materials 3472. Upon initialization, the operational materials 3472 validate the embedded signature value against the actual electronic appliance 600 signature SIG, and may refuse to start if the comparison fails.</p> <p>z. Depending on the configuration of electronic appliance 600, the machine signature may consist, for example, of some combination of a hash of the ROM BIOS 658' (see FIG. 69G), a hash of a disk defect map 3497a, the Ethernet (or other) network adapter 666 address, information written into an unused disk sector, information stored in a non-volatile CMOS RAM (such as used for hardware configuration data), information stored in non-volatile ('flash') memory (such as used for system or peripheral component 'BIOS' programs) and/or hidden unique information placed into the root directory 3497b of the fixed disk drive 668.</p> |



|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>aa. FIG. 69G shows an example of some of these appliance-specific signatures.</p> <p>bb. In this example, machine signature information need not be particularly large. Security is provided by hiding the machine signature rather than on any other cryptographic strength, because there is no more secure mechanism for key storage to protect it. Thus, it is satisfactory for the signature to be just large enough (e.g., two bytes) that it is unlikely to be duplicated by chance.</p> <p>cc. For some electronic appliances 600 where it can be determined that the technique is safe, an otherwise unused section of the non-volatile CMOS RAM 656a may be used to store a signature 3497d. Signature 3497d is verified against the PPE 650's internal state whenever the PPE is initialized. Signature 3497d may also be updated whenever a significant change is made to the secure database 610. If the CMOS RAM signature 3497d does not match the database value, PPE 650 may take this mismatch as an indication that a previous instance of the secure database 610 and/or PPE 650 software has been restored, and appropriate action can be taken. This mechanism thus ensures that even a bit-for-bit copy of the system's fixed disk 668 or other storage medium cannot be saved and reloaded to restore an earlier PPE 650 state. This particular technique depends upon there being an unused location available within CMOS RAM 656a, and may also require the CMOS RAM checksum algorithm to be known. An incorrect implementation could cause a subsequent reboot of electronic appliance 600 to fail because of a bad CMOS checksum, or worse, could alter some critical configuration parameter within CMOS RAM 656a so that electronic appliance 600 could not be recovered. Thus, care must be taken before modifying the contents of CMOS RAM 656a.</p> <p>dd. A still alternate technique may involve marking otherwise 'good' disk sectors 3497c defective and using the sector(s) to store machine signatures and/or encryption keys. This technique ensures that a logical bit-for-bit copy of the media does not result in a usable PPE 650 instance, and also provides relatively inaccessible and non-volatile storage for the information. Because a relatively large amount of storage space can be reserved using this technique, there is enough storage for a cryptographically strong value.</p> <p>ee. Some of the 'machine signature' techniques discussed above may be problematic in some electronic appliances 600 because it may be difficult to locate appropriate appliance-unique information. For example, although in a personal computer a ROM BIOS 658' is always available, the ROM BIOS information by itself may be insufficient because it is likely to be identical for a batch of electronic appliances 600 purchased together. Identifying a network adapter 666 and determining its address is potentially difficult due to the wide variety of adapters; additionally, an electronic appliance's network address may change (although this occurrence may be infrequent). Inserting random signature values into unused bytes within the fixed disk root directory 3497b and/or partition</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction   |
|--|----------------------|---|
|  |                      | <p>records may trigger some virus-checking programs, and the data may be modified by defragmentation or other disk manipulation programs. Where supported, a truly unused disk sector 3497c (e.g., one that is marked 'bad' even though it may still viably store information) may be used to store the machine signature. Even so, normal maintenance, upgrades or other failure recovery procedures may disrupt a particular machine association. Since the VDE administrator 200h participates in restoring a PPE 650 based on an encrypted backup image (as described above for example in connection with FIGS. 39-40), the VDE administrator may establish new associations at this point to maintain correspondence between a particular PPE 650 installation and a particular electronic appliance 600.</p> <p>ff. Tie Installation to Payment Method</p> <p>gg. A still additional example technique for providing additional security is to tie a particular PPE 650 installation at registration time to a particular payment method (see FIG. 69C, block 3470(8)). The registration process at installation time may thus serve to tie the PPE 650 installation to some payment method associated with the user, and to store the payment association information both within the PPE 650 instance and at the registry 3476. This technique assures that the actions of a particular PPE 650 instance are accountable to the assigned user with at least the reliability of whatever payment/credit verification technique is employed.</p> <p>hh. Install Operational Materials in Encrypted Form</p> <p>ii. Operational materials 3472 may first be customized as described above for the particular instance and/or appliance 600, then (at least mostly) encrypted for installation into the appliance such as by storage onto disk 668 (see FIG. 69C, block 3470(9)). Different installations may use different sets of decryption keys to decrypt the information once installed. Different parts of operational materials 3472 may be encrypted with different cryptographic keys to further complicate the analysis. This encryption makes analysis of the on disk form of the operational materials 3472 more difficult or infeasible.</p> <p>jj. The beginning of the resulting stored executable file may contain a small decryption program ('decryptor') that decrypts the remainder of the operational materials 3472 as they are loaded into memory. Confounding algorithms (as described below) may be used in this decryptor to make static recovery of the cryptographic keys difficult. Although the decryptor is necessarily in unencrypted form in an all-software installation without hardware support, the use of confounding algorithms to develop the associated cryptographic keys effectively requires a memory image to be captured after the program has been decrypted. Where supported (as described above), an unused and inaccessible disk sector 3497c may be used to store the decryption keys, and the operational materials 3472 may possess only the address for that particular sector. Embedding this address further complicates analysis.</p> <p>kk. Customized Layout</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>ll. The installation materials 3470 may store the encrypted operational materials 3472 onto the fixed disk 668 using a customized storage layout (FIG. 69C, block 3470(10)). FIG. 69F, 69H, 69I and 69J shows example customized software and data layouts. In these examples, each installed instance of operational materials 3472 is different in both executable form and in data layout. These modifications make each PPE 650 instance require separate analysis in order to determine the storage locations of its critical data such as cryptographic keys. This technique is an effective counter to creation of programs that can undo the protections of an arbitrary PPE 650 instance.</p> <p>mm. Instruction sequences within the operational materials 3472 may be modified by the installation routine to change the execution flow of the executable operational materials 3472 and to alter the locations at which the software expects to locate critical data. The alterations in program flow may include customization of time-consuming confounding algorithms. The locations of the modifiable instruction sequences may be embedded within operational materials 3470, and may therefore be not directly available from an examination of the installation and/or operational materials.</p> <p>nn. FIG. 69H shows one example operational materials 3472 executable code segment provided distinct processes 3498a, 3498b, 3498c, 3498d, 3498e. In this particular example, segment 3498a is executed first and segment 3498e is executed last, but the processes 3498b, 3498c and 3498d may be performed in any order (i.e., they are sequence independent processes). The installation materials 3470 may take advantage of this sequence independence by storing and/or executing them in different and/or depending upon the particular PPE instance 650. FIG. 69I, for example, shows a first static layout order, and FIG. 69J shows a second, different static layout order. Data elements associated with the executables may similarly be stored in different orders (as shown in FIGS. 69I, 69J) depending upon the particular installation.</p> <p>oo. Dynamic Protection Mechanisms</p> <p>pp. In addition to the more static protection mechanisms described above, dynamic protection mechanisms may be employed to complicate both static and dynamic analysis of the executable (executing) operational materials 3472. Such techniques include, for example:</p> <p>qq. implementation complexity, immediate overwriting, hardware dependent sequences, timing dependencies, confounding algorithms, random modifications, dynamic load module decryption,</p> <p>rr. on-line integrity checks, time integrity checks, machine association integrity checks, dynamic storage integrity checks, and hidden secret storage volatile secret storage internal consistency checks.</p> <p>ss. FIGS. 69K-69L show an example execution of operational materials 3472 that may employ some or all of these various dynamic protection mechanisms.</p> <p>tt. Upon starting execution (FIG. 69K, block 3550), the installed operational</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>materials 3472 may run initialization code as described above that is used to decrypt the stored encrypted operational materials on an 'as needed' basis (FIG. 69K, block 3552). This initialization code may also check the current value of the real-time clock (FIG. 69K, block 3554).</p> <p>uu. Real Time Check/Validation</p> <p>vv. Operational materials 3472 may perform this time check, for example, to guard against replay attacks and to ensure that the electronic appliance 600's time is in reasonable agreement with that of the VDE administrator 200h or other trusted node.</p> <p>ww. FIG. 69M shows an example sequence of steps that may be performed by the 'check time' block 3554. In this example, PPE 650 uses secure communications (e.g. a cryptographic protocol) to obtain the current real time from a trusted server (FIG. 69M, block 3554a). PPE 650 may next ask the user if he or she wishes to reset the electronic appliance real-time clock 528 (which may, for example, be the real-time clock module within a personal computer or the like) so it is synchronized with the trusted server's time clock.</p> <p>xx. If the user responds affirmatively, PPE 650 may reset the time clock to agree with the real-time provided by the trusted server ('yes' exit to decision block 3554b, FIG. 69M, block 3554c). If the user responds that he or she does not want the real-time clock reset ('no' exit to decision block 3554b), then PPE 650 may calculate a delta value of the difference between the server's real-time clock and the electronic appliance's real-time clock 528 (FIG. 69M, block 3554d). In either case, PPE 650 may store the current time T<sub>current</sub> into a non-volatile storage location T<sub>store</sub> indicating the current real-time (FIG. 69M, block 3554e).</p> <p>yy. Referring again to FIG. 69K, PPE 650 can disable itself if there is too much (or the wrong type) of a difference between the trusted server's time and the electronic appliance's clock—since such differences can indicate replay attacks, the possibility that the PPE 650 has been restored based on a previous state, etc. For example, if desired, PPE 650 can generate a time check fail exception if the electronic appliance's real-time clock 528 disagrees with the trusted server's real-time by more than a certain amount of acceptable drift (FIG. 69K, 'yes' exit to decision block 3556). In the event of such an exception, PPE 650 may disable itself (FIG. 69K, block 3558) and require a dialog between the user and registry 3476 (or other authority)—providing additional protection against replay attacks and also detecting clock failures that could lead to incorrect operation or incorrect charges.</p> <p>zz. Dynamic Code Decryption and Data OverWriting</p> <p>aaa. Operational materials 3472 may then decrypt the next program segment dynamically (FIG. 69K, block 3460. The code may be decrypted dynamically when it is needed, then re-encrypted or overwritten and discarded when not in use. This mechanism increases the tamper-resistance of the executable code—thus providing additional tamper resistance for PPE operations. As mentioned above, different decryption</p> |

|  | Claim<br>Term/Phrase | Evidence Supporting MS Construction  |
|--|----------------------|--|
|  |                      | <p>keys may be required to decode different code portions, and the decryption keys can be installation-specific so that an attacker who successfully comprises the decryption key of one instance cannot use that information to compromise any other instance's decryption key(s).</p> <p>bbb. Once a portion of the operational materials 3472 has been decrypted (FIG. 69K, block 3560), that portion may immediately overwrite all initialization code in memory since it is no longer required (FIG. 69K, block 3562). The executing operational materials 3472 may similarly overwrite all unwrapped cryptographic keys once they are no longer needed, and may also overwrite expanded key information developed by initializing the cryptographic algorithms once no longer needed. These techniques minimize the amount of time during which usable key information is available for exposure in a memory snapshot—complicating all but the most dynamic of analysis efforts. Because all keys in permanent storage are either encrypted or otherwise camouflaged, no such treatment is required for I/O buffers.</p> <p>ccc. Dynamic Check of Association Between Appliance and PPE Instance</p> <p>ddd. The executing operational materials 3472 may next compare an embedded electronic appliance signature SIG' against the electronic appliance signature SIG stored in the electronic appliance itself (FIG. 69K, decision block 3564). As discussed above, this technique may be used to help prevent operational materials 3472 from operating on any electronic appliance 600 other than the one it was initially installed on. PPE 650 may disable operation if this machine signature check fails ('no' exit to decision block 3564, FIG. 69K; disable block 3566).</p> <p>eee. Self-Modifying and/or Hardware-Dependent Code Sequences</p> <p>fff. Executing operational materials 3472 may also employ self-modifying code sequences that cannot easily be emulated with a software debugger or single-stepping program (FIG. 69K, block 3568). These sequences may, for example, be dependent on specific models of electronic appliances 600, and may be patched into the operational materials 3472 as appropriate to installation materials 3470 based on tests performed during the installation process. Such hardware-dependent sequences may be used to ensure that critical algorithms yield different results when executed on the proper hardware as opposed to when executed on different hardware or under software control such as in a debugger or emulator. To prevent such hardware-dependent sequences from being readily recognizable from a static examination of the code, the sequences may be constructed at run time and then invoked so that they can be identified only by analysis of the instruction sequences actually executed.</p> <p>ggg. Dynamic Timing Checks</p> <p>hhh. Executing operational materials 3472 may also make dynamic timing checks on various code sequences, and refuse to operate if they do not execute within the expected interval (FIG. 69K, block 3570, decision block 3572, 'disable' block 3574). An incorrect execution time suggests that the operational materials 3472 are being externally manipulated</p> |

|     | Claim<br>Term/Phrase   | Evidence Supporting MS Construction   |
|-----|--|---|
|     |  | <p>and/or analyzed or traced in some manner (e.g., by a software emulator). This technique thus provides additional protection against dynamic analysis and/or modification.</p> <p>iii. The expected execution intervals associated with certain code sequences may be calculated during the installation procedure. Resulting test values may be embedded into the operational materials 3472. These timing tests may be integrated with time integrity tests and dynamic integrity checks to make it more difficult to bypass them simply by patching out the timing check. Care should be taken to eliminate false alarms due to concurrent system activity (e.g., other tasks and/or windows).” (‘900 235:28 - 244:15)</p> <p>4. See also ‘900 Figs. 69A-N</p> |
| 30. | 912.8:<br>“identifying at least one aspect of an execution space required for use and/or execution of the load module” | <p><b>Intrinsic:</b></p> <p>1. “For each site, the manufacturer generates a site ID 2821 and list of site characteristics 2822.” (‘193 209:55-57)</p>   |

**Appendix 1 to Exhibit D: Source Abbreviations**

**Intrinsic Evidence:**

| <b>Abbreviated Reference</b> | <b>Full Citation or Title</b>                 |
|------------------------------|---|
| '193                         | U.S. Patent No. 6,253,193                     |
| '683                         | U.S. Patent No. 6,185,683                     |
| '721                         | U.S. Patent No. 6,157,721                     |
| '861                         | U.S. Patent No. 5,920,861                     |
| '891                         | U.S. Patent No. 5,982,891                     |
| '900                         | U.S. Patent No. 5,892,900                     |
| '912                         | U.S. Patent No. 5,917,912                     |
| '712                         | U.S. Patent Application Serial No. 08/699,712 |
| '107                         | U.S. Patent Application Serial No. 08/388,107 |

**Extrinsic Evidence:**

| <b>Abbreviated Reference</b> | <b>Full Citation or Title</b>   |
|------------------------------|---|
| Bishop                       | M. Bishop, <u>Computer Security, Art &amp; Science</u> , (2003).  |
| Booth                        | C. J. Booth, ed. <u>The New IEEE Standard Dictionary of Electrical and Electronics Terms</u> , 5 <sup>th</sup> edition, (1993). |
| Davies                       | D.W. Davies and W.L. Price, <u>Security for Computer Networks</u> , (1984) MSI083423-MIS083443.                                 |
| Denning                      | D. Denning, <u>Cryptography and Data Security</u> , (1983), MSI085569.  |
| Dictionary of Computing      | <u>Dictionary of Computing</u> , 3 <sup>rd</sup> edition, Oxford University Press, (1990).                                      |
| IBM                          | G. McDaniel, ed., <u>IBM Dictionary of Computing</u> , (1994).  |
| Laplante                     | P. A. Laplante, ed., <u>Dictionary of Computer Science, Engineering, and Technology</u> (2001).                                 |
| Longley                      | D. Longley, et al., <u>Information Security: Dictionary of Concepts, Standards and Terms</u> , (1992).                          |
| Neumann                      | P.G. Neumann, <u>Computer Related Risks</u> , (1995).   |
| Pfleeger                     | C. P. Pfleeger, <u>Security in Computing</u> , (1989).  |
| Que                          | C. Weisert, <u>Que's Computer Programmer's Dictionary</u> , (1993).   |
| Russell                      | D. Russell and G.T. Gangemi, <u>Computer Security Basics</u> , (1991).  |
| Webster's                    | D. Spencer, <u>Webster's New World Dictionary of Computer Terms</u> , 4 <sup>th</sup> edition (1992).                           |

## Exhibit E

### Microsoft's Statement of Reservations

Microsoft provides its attached claim construction for each of the 30 "Mini-Markman" terms and phrases, subject to the limitations and reservations of rights set forth herein.

**Claim Invalidity:** Microsoft does not waive any defenses that the asserted claims fail to satisfy the provisions of 35 U.S.C. § 112, including, for example, the written description requirement, the definiteness requirement, or any other requirement for patentability. Microsoft does not concede that the asserted claims are supported by Plaintiffs original "big book" application or any application from which they purportedly claim priority. By offering a construction of a term, Microsoft does not waive any defense that the claim is indefinite and there can be no proper construction.

**Continuing Discovery:** Microsoft reserves the right to modify its claim constructions in light of ongoing claim construction discovery, in particular such discovery compelled by Judge James' Order of March 10, 2003. Microsoft reserves the right to modify or supplement its cited extrinsic evidence in light of information that is provided in continuing discovery on claim construction and indefiniteness.

**Intrinsic Evidence:** For the purposes of submission of this claim construction only, Microsoft treats the "intrinsic" evidence as including: 1) the specifications of each of the seven U.S. patents at issue in the "Mini-Markman" proceeding, including any material purportedly incorporated by reference therein; 2) the prosecution history of each of the seven patents at issue, including the applications and prosecution history of the seven patents and any related patent applications, including without limitation, applications purportedly incorporated by reference or to which an application claimed priority; and 3) all references cited in the prosecution of any such applications. Microsoft does so without waiving the right to contest whether some of this information is or is not properly part of the intrinsic evidence.



Dr. Reiter is expected to testify as follows:

1. Dr. Reiter will testify regarding the meaning of the disputed claim elements to one of ordinary skill in the art, taking into account the understood meaning of the terms in the art, the patent specifications and the file histories. He will testify as follows:

a. InterTrust's proposed definitions, attached as Exhibit B to the Joint Claim Construction Statement ("JCCS") are consistent with the use of the terms or phrases in the specification and the relevant art. Those definitions are attached hereto. Citations to supporting specification text and relevant art can be found in Exhibit C to the JCCS.

b. Microsoft has made repeated substantial changes to its proposed definitions, the changes continuing up to shortly before the present document was prepared. For this reason, it is impossible to include detailed responses to the issues raised by those definitions.

In general, however, the Microsoft definitions incorporate restrictions that are inconsistent with specification use of the terms and/or inconsistent with the understanding of the terms in the art. Those inconsistencies are demonstrated by the attached supporting evidence. The following discussion lists one or more serious deficiencies in each Microsoft definition, but is not intended as a comprehensive description of all such deficiencies.

#### **Individual terms**

##### **Access/Access to/Accessing/Accessed**

The first sentence of Microsoft's definition is generally consistent with the InterTrust definition. The second sentence of the Microsoft definition is based on a specific disclosed embodiment, and is inconsistent with general use of the term in the specifications.

##### **Addressing**

The two parties' definitions are very close. Microsoft's definition is, however, improper in its apparent exclusion of indirect addressing.

##### **Allowing, allows**

Microsoft's definition is based on a specific disclosed embodiment and ignores other embodiments. See InterTrust's supporting evidence.

##### **Arrangement**

Microsoft's definition requires particular types of organizations and is therefore inconsistent with the patent specifications.

#### **Aspect**

Microsoft's definition is overly restrictive in its requirement that an aspect be "persistent" and that it "can be used to distinguish [an environment] from other environments."

#### **Associated with**

Microsoft's definition incorporates restrictions based on a particular embodiment and is inconsistent with other disclosed embodiments and with the general meaning of the term.

#### **Authentication**

Microsoft's definition requires multiple types of authentication, in a manner not required by use of this term in the specification or the art. Moreover, some of these types cannot be applied (e.g., "origin integrity" applied to an organization).

#### **Authorization information, Authorized, Not authorized**

Microsoft's definitions are based on specific embodiments and contradicted by alternative embodiments disclosed in the specifications.

#### **Budget control; Budget**

Microsoft's definition improperly restricts "budget" to a particular type of method, and improperly restricts Budget Control in a manner inconsistent with the specification.

#### **Can be**

Microsoft's definition incorporates the language "which otherwise cannot be carried out." This language is inconsistent with the specifications.

#### **Capacity**

The Microsoft definition relates to hardware storage devices, a context that is irrelevant to use of the term in the relevant claim.

#### **Clearinghouse**

Microsoft's definition is inconsistent with use of this term in the specifications. See InterTrust's supporting evidence.

**Compares; Comparison**

Microsoft's definition is based on a particular type of processor operation, a context that is not discussed in the specification and not required by the claim.

**Component assembly**

Microsoft's definition incorporates a large number of restrictions based on specific embodiments and ignoring alternate embodiments.

**Contain, contained, containing**

Microsoft's definition requires "physically" or "directly" storing, and distinguishes Addressing. This is inconsistent with use of the term in the specification.

**Control (n.); Controls (n.)**

The Microsoft definition incorporates a large number of restrictions based on specific embodiments, and ignores alternate embodiments described in the specifications.

**Controlling; Control (v.)**

The Microsoft definition incorporates limitations that are not required by the specification, including limitations contradicted by use of the term in the specifications and by disclosed embodiments.

**Copied file**

The Microsoft definition improperly distinguishes "copied file" from "copy."

**Copy, copied, copying (v.)**

The Microsoft definition is internally inconsistent, since it both prohibits and allows changes in the reproduced file. That definition also incorporates examples that are inconsistent with use of the terms in the claims.

**Copy control**

The Microsoft definition is inconsistent with use of this term in the claim.

**Data item**

The Microsoft definition incorporates limitations not present in the InterTrust definition. These limitations are not required by the specification or normal use of the term in the art.

### **Derive, Derives**

The Microsoft definition requires retrieval, a concept not required by the specifications or use of this term in the claim.

### **Descriptive data structure**

Limitations in the last two sentences of the Microsoft definition are inconsistent with described embodiments and are not required by the specifications or use of the term in the claims.

### **Designating**

The Microsoft definition does not apply to this term, but instead to the claim phrase in which the term is found. That claim phrase is separately defined.

### **Device class**

The Microsoft definition is inconsistent with the definition given to this term during prosecution.

### **Digital file**

The Microsoft definition is overly restrictive. The limitations is incorporates are not required by the specification, use of the term in the claims or general use in the relevant art.

### **Digital signature; Digitally signing**

The Microsoft definition of digital signature requires that the string be "computationally unforgeable," a characteristic that is impossible to obtain. The Microsoft definition of digitally signing requires a secret key, and also includes significant background discussion not necessary for the definition.

### **Entity's control**

Microsoft's definition improperly requires control of a "particular use of or access to particular protected information by a particular user(s)." No such requirements are imposed by the term, the claim or the specifications.

### **Environment**

Microsoft does not appear to have provided any definition for this term.

### **Executable programming; Executable**

Microsoft's requirement of "machine code instructions" is inconsistent with use of this term in the specifications. In addition, Microsoft's definition of "computer program" imposes limitations not required by these terms.

#### **Execution space; Execution space identifier**

Microsoft's definition of Execution Space is inconsistent with the explicit definition given to this term during prosecution. Microsoft's definition of Execution Space Identifier improperly requires "unique" identification.

#### **Governed item**

Microsoft's definition of Governed Item requires arbitrarily fine granularity and control of "access and use by any user, process, or device." Neither the term nor the specifications require such limitations.

#### **Halting**

The Microsoft definition requires execution be "unconditionally" stopped. The specification imposes no such requirement, and the Microsoft definition appears to be based on a particular type of instruction that is not mentioned in the patents.

#### **Host processing environment**

The Microsoft definition incorporates the term "VDE node," a term that is itself defined at great length, incorporating numerous improper limitations. The Microsoft definition also improperly incorporates restrictions based on privileged mode versus user mode, and "loaded" software. In addition, the Microsoft definition improperly excludes hardware.

#### **Identifier, Identify, Identifying**

The Microsoft definitions improperly restrict these terms to "particular instances."

#### **Including**

The definitions are consistent, except that the hardware portion of Microsoft's definition requires "physically present within." This is inconsistent with use of the term in the claims.

#### **Information previously stored**

Microsoft's definition would render the claim nonsensical, since it would require a comparison involving information that is no longer available for the comparison.

### **Integrity programming**

The Microsoft definition is internally inconsistent, improperly incorporates the term Executable Programming and improperly defines integrity as excluding all alterations.

### **Key**

Microsoft's exclusion of "key seed or other information from which the actual encryption and/or decryption key is constructed, derived, or otherwise identified" is inconsistent with the specification and general use of the term in the relevant art.

### **Load module**

Microsoft's definition imposes numerous limitations beyond those identified in the InterTrust definition. Those additional limitations are not required by the term and are inconsistent with embodiments disclosed in the specifications.

### **Machine check programming**

The Microsoft definition improperly requires Executable Programming and a "unique 'machine signature' which distinguishes the physical machine from all other machines." These limitations are not required by the term.

### **Opening secure containers**

The Microsoft definition improperly distinguishes "opening" from decrypting, and improperly incorporates limitations based on a particular embodiment of opening.

### **Operating environment**

See Processing Environment.

### **Organization, Organization information, Organize**

The Microsoft definitions improperly incorporate concepts related to physical storage.

### **Portion**

The Microsoft definition improperly implies that presence of a "portion" excludes presence of the whole.

### **Prevents**

The Microsoft definition requires a level of certainty that is inconsistent with the specification and impossible to obtain.

#### **Processing Environment**

The Microsoft definition incorporates a specific embodiment and would exclude other embodiments disclosed for this term.

#### **Protected processing environment**

The Microsoft definition incorporates at least several dozen highly restrictive and unnecessary limitations, and appears to combine restrictions from multiple separate embodiments.

#### **Protecting**

The incorporation of Security into the Microsoft definition is improper, since that term is considerably more general than the manner in which Protecting is used in the claim.

#### **Record**

The Microsoft definition includes limitations beyond those incorporated in the InterTrust definition. These added limitations are not required by use of this term in the claims, specification, or art.

#### **Required**

The Microsoft definition implies a degree of absoluteness that is inconsistent with the specification. The second sentence of the Microsoft definition is unsupported by the specification or normal use of the term.

#### **Resource processed**

The Microsoft definition improperly requires a "shared facility," and that the resource be "required by a job or task." These are not required by the claim or specification.

#### **Rule**

The Microsoft definition improperly distinguishes Rules from Controls, and imposes an unsupported requirement that a Rule be a "lexical statement."

#### **Secure**

The Microsoft definition requires absolute protection against all possible threats, and is therefore inconsistent with use of the term in the specification, the claims, and the relevant art.

#### **Secure container**

The requirements imposed by the Microsoft definition are either inconsistent with the specification or ignore disclosed embodiments.

#### **Secure container governed item**

The Microsoft definition imposes a requirement of absolute security that is inconsistent with the specification and ignores alternate disclosed embodiments.

#### **Secure database**

The Microsoft definition improperly defines "database" in accordance with one particular type of database, and improperly imposes a requirement of absolute security that is inconsistent with the specification.

#### **Secure execution space**

The Microsoft definition is inconsistent with and excludes embodiments of Secure Execution Spaces described in the specification.

#### **Secure memory**

Microsoft's definition of "memory" improperly excludes virtual memory. Microsoft's definition of Secure Memory includes numerous restrictions not supported by the specification.

#### **Secure operating environment, Said operating environment**

See Secure Processing Environment.

#### **Securely applying**

Microsoft's definition of "securely" is inconsistent with and excludes embodiments described in the specification.

Microsoft's definition of Securely Applying improperly includes limitations from specific embodiments, as well as limitations not required by the specification or claims.

#### **Securely assembling**



The Microsoft definition incorporates limitations from specific embodiments, and ignores alternate embodiments not requiring those limitations.

#### **Securely processing**

The Microsoft definition improperly incorporates a requirement of a secure execution space. This requirement is inconsistent with embodiments described in the specification.

#### **Securely receiving**

The Microsoft definition is based on limitations taken from a particular embodiment and ignores alternate embodiments.

#### **Security level, Level of security**

The Microsoft definition improperly requires an "ordered measure" and persistence. The second and third sentences from the Microsoft definition are unsupported by any disclosure in the specifications.

#### **Tamper resistance**

The Microsoft definition improperly requires a tamper resistant barrier.

#### **Tamper resistant barrier**

The Microsoft definition describes a specific embodiment, and is inconsistent with alternate embodiments described in the specifications.

#### **Tamper resistant software**

The Microsoft definition improperly requires a tamper resistant barrier.

#### **Use**

The second sentence of the Microsoft definition improperly incorporates limitations from a particular embodiment.

#### **User controls**

The Microsoft definition is inconsistent with the claim and the prosecution history.

#### **Validity**

The Microsoft definition improperly incorporates the concept of “authentication,” and applies only to data.

**Virtual distribution environment**

See Global Construction of VDE.

**Claim phrases**

**193.1**

**receiving a digital file including music**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication a recipient and use of controls.

**a budget specifying the number of copies which can be made of said digital file**

The Microsoft definition improperly includes “copies” that are not “long-lived, decrypted or accessible.” The Microsoft definition also ignores embodiments involving alternative control structures.

**controlling the copies made of said digital file**

The Microsoft definition improperly incorporates limitations from particular embodiments, ignores embodiments describing alternative control structures and imposes numerous limitations that are not supported by the specification or claim language.

**determining whether said digital file may be copied and stored on a second device based on at least said copy control**

The Microsoft definition incorporates numerous unnecessary limitations not required by the claim or the specification, improperly requires that “the” file, as opposed to a copy, be stored on a second device, excludes described alternative embodiments and requires an absolute degree of control that is inconsistent with the specification.

**if said copy control allows at least a portion of said digital file to be copied and stored on a second device**

The Microsoft definition’s “explanation” of the branches makes no sense and is unsupported by the claim and , improperly requires that “the” file, as opposed to a copy, be stored on a second device.

**copying at least a portion of said digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly excludes embodiments described in the specification.

**transferring at least a portion of said digital file to a second device**

The Microsoft definition improperly distinguishes a "copy" and "the" file, improperly requires that controls be executed and ignores alternative embodiments described in the specification.

**storing said digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly requires storage of the entire file rather than a portion.

**193.11**

**receiving a digital file**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication a recipient and use of controls.

**determining whether said digital file may be copied and stored on a second device based on said first control**

The Microsoft definition incorporates numerous unnecessary limitations not required by the claim or the specification, improperly requires that "the" file, as opposed to a copy, be stored on a second device, excludes described alternative embodiments and requires an absolute degree of control that is inconsistent with the specification.

**identifying said second device**

The Microsoft definition improperly requires that the identification distinguish the device from all other devices, that controls be used and that a VDE Secure Processing Environment be used.

**whether said first control allows transfer of said copied file to said second device**

The Microsoft definition improperly distinguishes a "copy" from "the" file, and ignores embodiments describing alternative control structures.

**said determination based at least in part on the features present at the device**

The Microsoft definition improperly requires that all features be used, that these be "actual, current" features and improperly excludes device identifiers.

**if said first control allows at least a portion of said digital file to be copied and stored on a second device**

The Microsoft definition's "explanation" of the branches makes no sense and is unsupported by the claim and , improperly requires that "the" file, as opposed to a copy, be stored on a second device.

**copying at least a portion of said digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly excludes embodiments described in the specification.

**transferring at least a portion of said digital file to a second device**

The Microsoft definition improperly distinguishes a "copy" and "the" file, improperly requires that controls be executed and ignores alternative embodiments described in the specification.

**storing said digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly requires storage of the entire file rather than a portion.

**193.15**

**receiving a digital file**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication a recipient and use of controls, and the requirement that the step must proceed in both authentication branches is not supported in the claim.

**an authentication step comprising:**

The Microsoft definition improperly includes a requirement of an absence of trust, VDE controls and a VDE Secure Processing Environment.

**accessing at least one identifier associated with a first device or with a user of said first device**

The Microsoft definition improperly requires "securely" accessing, that an identifier identify a "single" user or device (but not "and"), VDE controls, and a VDE Secure Processing Environment.

**determining whether said identifier is associated with a device and/or user authorized to store said digital file**

The Microsoft definition improperly requires VDE controls and a VDE Secure Processing Environment.

**storing said digital file in a first secure memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized**

The Microsoft definition ignores embodiments describing alternative control structures, and improperly requires that “the” file be stored, as opposed to a copy, VDE controls, and a VDE Secure Processing Environment.

**storing information associated with said digital file in a secure database stored on said first device, said information including at least one control**

Microsoft’s definition improperly requires that the stored information be associated with the digital file but not the digital file’s contents, VDE controls, a VDE Secure Processing Environment and that the step proceed regardless of the outcome of the authentication step.

**determining whether said digital file may be copied and stored on a second device based on said at least one control**

The Microsoft definition incorporates numerous unnecessary limitations not required by the claim or the specification, improperly requires that “the” file, as opposed to a copy, be stored on a second device, excludes described alternative embodiments, requires an absolute degree of control that is inconsistent with the specification, and requires that the step proceed regardless of the outcome of the authentication step.

**if said at least one control allows at least a portion of said digital file to be copied and stored on a second device,**

The Microsoft definition’s “explanation” of the branches makes no sense and is unsupported by the claim and , improperly requires that “the” file, as opposed to a copy, be stored on a second device.

**copying at least a portion of said digital file**

The Microsoft definition improperly distinguishes a “copy” and “the” file, and improperly excludes embodiments described in the specification and improperly requires that the step proceed regardless of the outcome of the authentication step.

**transferring at least a portion of said digital file to a second device**

The Microsoft definition improperly distinguishes a “copy” and “the” file, improperly requires that controls be executed and ignores alternative embodiments

described in the specification, and improperly requires that the step proceed regardless of the outcome of the authentication step.

**storing said digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly requires storage of the entire file rather than a portion, and improperly requires that the step proceed regardless of the outcome of the authentication step.

**193.19**

**receiving a digital file at a first device**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication a recipient and use of controls.

**establishing communication between said first device and a clearinghouse located at a location remote from said first device**

The Microsoft definition improperly requires a communications channel and that the communications channel was "previously non-existent."

**using said authorization information to gain access to or make at least one use of said first digital file**

The Microsoft definition improperly requires that "all of" the authorization information be used, VDE controls, a VDE Secure Processing Environment, and ignores embodiments describing alternative control structures.

**receiving a first control from said clearinghouse at said first device**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication a recipient and use of controls.

**storing said first digital file in a memory of said first device**

The Microsoft definition improperly requires VDE controls and a VDE Secure Processing Environment.

**using said first control to determine whether said first digital file may be copied and stored on a second device**

The Microsoft definition incorporates numerous unnecessary limitations not required by the claim or the specification, improperly requires that "the" file, as opposed

to a copy, be stored on a second device, excludes described alternative embodiments and requires an absolute degree of control that is inconsistent with the specification.

**if said first control allows at least a portion of said first digital file to be copied and stored on a second device**

The Microsoft definition's "explanation" of the branches makes no sense and is unsupported by the claim and , improperly requires that "the" file, as opposed to a copy, be stored on a second device.

**copying at least a portion of said first digital file**

The Microsoft definition improperly distinguishes a "copy" and "the" file, and improperly excludes embodiments described in the specification.

**transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output**

The Microsoft definition improperly distinguishes a "copy" and "the" file, improperly requires that controls be executed and ignores alternative embodiments described in the specification.

**storing said first digital file portion**

Microsoft's definition improperly distinguishes a "copy" and "the" file.

**683.2**

**the first secure container having been received from a second apparatus**

Microsoft's definition improperly requires that the first secure container identify the apparatus from which it was received, and improperly argues that, in the absence of such identification, that container could not be distinguished from a container created at the site. Microsoft's definition includes numerous improper limitations, including authenticating a recipient and authentication occurring in accordance with VDE controls. The examples cited by Microsoft are misleading, since these are specific embodiments rather than general requirements.

**an aspect of access to or use of**

Microsoft's definition improperly excludes rules governing more than one aspect, improperly excludes access and use and improperly requires that the aspect be governed in relation to "any and all processes, users, and devices."

**the first secure container rule having been received from a third apparatus different from said second apparatus**

Microsoft's definition improperly requires that the first secure container identify the apparatus from which it was received, and improperly argues that, in the absence of such identification, that container could not be distinguished from a container created at the site. Microsoft's definition includes numerous improper limitations, including receipt in a secure container, authenticating a recipient and authentication occurring in accordance with VDE controls.

**hardware or software used for receiving and opening secure containers**

Microsoft's definition improperly requires a Secure Processing Environment and SPU, improperly requires "the same single logical piece of either hardware or software (as opposed to both)," and improperly requires authentication and VDE controls.

**said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers**

The Microsoft definition improperly requires that rules be associated with secure containers, as opposed to governed items.

**protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus**

The Microsoft definition is unsupported in the specification. It is contradicted by the claim and improperly requires numerous elements not required by the specification, including a Secure Processing Environment.

**hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container**

The Microsoft definition improperly requires a Secure Processing Environment/SPU, a "single" piece of hardware or software, assembly of a control and governance through VDE controls.

**hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.**

The Microsoft definition improperly requires a Secure Processing Environment/SPU, a "single" piece of hardware or software, assembly of a control and governance through VDE controls. The examples cited by Microsoft are misleading, since these are specific embodiments rather than general requirements.



**digitally signing a first load module with a first digital signature designating the first load module for use by a first device class**

The Microsoft definition improperly requires that the digital signature be used as the signature key, that all load modules be signed and that certain devices not have keys.

**digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class**

The Microsoft definition improperly requires that the digital signature be used as the signature key, that all load modules be signed, that certain devices not have keys, that security levels be persistent and that security levels be greater or less than other security levels.

**distributing the first load module for use by at least one device in the first device class**

The Microsoft definition improperly requires transmission and that the digital signature accompany the first load module as distributed.

**distributing the second load module for use by at least one device in the second device class**

The Microsoft definition improperly requires transmission and that the digital signature accompany the first load module as distributed.

721.34

**arrangement within the first tamper resistant barrier**

The Microsoft definition improperly requires that the arrangement be "executed wholly within the first tamper resistant barrier."

**prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level**

The Microsoft definition improperly requires that the second secure execution space be part of the protected processing environment, that security level differences be persistent and higher or lower than each other and that the "same" executable be executed.

861.58

**creating a first secure container**

The Microsoft definition improperly requires a VDE Secure Processing Environment.

**including or addressing . . . organization information . . . desired organization of a content section. . . and metadata information at least in part specifying at least one step required or desired in creation of said first secure container**

The second paragraph from Microsoft's definition is inconsistent with the claim. The limitations imposed by the third paragraph are not required by the claim or specification.

**at least in part determine specific information required to be included in said first secure container contents**

The Microsoft definition improperly excludes other reasons for inclusion of the information and improperly requires specific values.

**rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents**

The Microsoft definition improperly requires that the rule be designed for particular contents, that the rule be used by VDE controls, the presence of a VDE Secure Processing Environment and that the rule is generated or identified based on the descriptive data structure. Microsoft's definition also excludes embodiments describing alternative control structures.

**891.1**

**resource processed in a secure operating environment at a first appliance**

The Microsoft definition improperly requires a shared facility and a Secure Processing Unit with specific features.

**securely receiving a first entity's control at said first appliance**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication, use of controls and encryption on the communications level.

**securely receiving a second entity's control at said first appliance**

The Microsoft definition includes numerous unnecessary limitations, including secure container, authentication, use of controls and encryption on the communications level.

**securely processing a data item at said first appliance, using at least one resource**

The Microsoft definition improperly requires a Secure Processing Unit including numerous limitations.

**securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item**

The Microsoft definition improperly requires a Secure Processing Environment consisting of a Secure Processing Unit and that the resource be a component part of a secure operating environment.

**900.155**

**first host processing environment comprising**

The Microsoft definition incorporates limitations not required by the claim or the specifications, including limiting the host processing environment to only currently executing software.

**designed to be loaded into said main memory and executed by said central processing unit**

The Microsoft definition improperly requires that the software is capable of being loaded "only" in the main memory and executed "only" by the CPU.

**said tamper resistant software comprising: . . . one or more storage locations storing said information**

The Microsoft definition improperly requires that the storage locations be part of the machine check programming and that the storage locations must not store other information.

**derives information from one or more aspects of said host processing environment,**

The Microsoft definition improperly requires that information be derived from "hardware," and that the information "uniquely and persistently" identify the host processing environment.

**one or more storage locations storing said information**

The Microsoft definition improperly requires that the storage locations be part of the tamper resistant software and that the storage locations must not store other information.

**information previously stored in said one or more storage locations**

Microsoft's definition would render the claim nonsensical, since it would require a comparison involving information that is no longer available for the comparison.

**generates an indication based on the result of said comparison**

Microsoft's definition improperly requires that only two results be possible and that the indication is based solely on the result of the "compares" step.

**programming which takes one or more actions based on the state of said indication**

The Microsoft definition improperly requires executable programming, that the programming not be part of the host processing environment, that the programming must take an action regardless of the indicator state and that the action must be based solely on the state of the indication.

**at least temporarily halting further processing**

Microsoft's definition improperly requires that the host processing environment and all processes running in it be halted.

912.8

**identifying at least one aspect of an execution space required for use and/or execution of the load module**

The Microsoft definition improperly requires that the identifier "define fully, without reference to any other information."

**said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security**

The Microsoft definition improperly requires that the execution space identifier provides the load module with the ability to determine a level of security, and the presence of two higher and two lower levels of security.

**checking said record for validity prior to performing said executing step**

The Microsoft definition improperly requires that the record be checked before execution of any identified information, that evaluation occur within a VDE Secure Processing Environment, and that specific types of information be checked.

**912.35**

**received in a secure container**

The Microsoft definition improperly requires "encapsulation" in a secure container, authentication in accordance with VDE controls and acceptance of the secured container.

**said component assembly allowing access to or use of specified information**

The Microsoft definition improperly requires that the component assembly operate by itself, that it execute in a VDE Secure Processing Environment and that the component assembly be dedicated to specific information. The Microsoft definition ignores embodiments describing alternative control structures and improperly distinguishes access and use.

**said first component assembly specified by said first record**

The first paragraph of Microsoft's definition defines this term in a restrictive manner with no support in the claim. Microsoft's second paragraph is devoted to a non-existent inconsistency created by Microsoft's restrictive definition.

**Claims as a Whole:**

In every case, Microsoft requires the system be a VDE or the method be performed in a VDE. This requirement is not supported by the language of any of the claims.

**Global Construction**

The language of the individual claims contains nothing to support the large number of restrictions imposed by Microsoft's "global construction." Those restrictions are unsupported by and in many cases contradicted by the specification.

2. Digital Rights Management in general. Dr. Reiter will testify regarding Digital Rights Management technology, including encryption and tamper-resistance techniques. The nature and extent of such testimony will depend on the Court's decision as to the scope and format of tutorial presentations.

3. InterTrust's patents and patent claims. Dr. Reiter will testify regarding the general nature of the InterTrust patents, and will summarize the claims at issue in the initial Joint Claim Construction hearing. The nature of that testimony will depend on the

Court's decision as to ordering and format of testimony, but will be consistent with the testimony outlined above regarding claim terms and phrases.

## **Exhibit G**

### **Summary of Opinions of Professor John Mitchell In Support of Microsoft's Proposed Claim Constructions**

1. In the field of computer security, terms such as "secure," "protect," and "tamper resistance" are understood differently depending on the particular context in which they are used. They have such a range of possible meanings that context is essential to understanding what these terms mean in a given instance. The same is true for terms like "govern" and "control" when they are used to describe computer systems or access to information.

A person skilled in the computer security field would not expect to use a dictionary to understand what these terms mean in a given context; rather, he or she would expect to review the particular reference or system in question to see what adversarial events or attacks are being defended against. Generally speaking, dictionary "definitions" are not sufficient for understanding how these terms are meant in a particular case. A number of terms and phrases used in the February 1995 application (such as "VDE," "PPE," and "secure container") are also not likely to be found in dictionaries.

2. The February 1995 application (which is sometimes referred to as the "Big Book") never clearly explains what it means by "security." It would not be clear to someone of average skill in the field what "secure" means in that application -- for example, with regard to systems, system components, information, or processes. The same is true for such terms as "protected" and "tamper resistant."

3. If a reasonably skillful computer security professional were to presume that "secure" has all of the attributes that are promised in the February 1995 application, then "secure" requires a guarantee of secrecy, authenticity, integrity, nonrepudiation, and availability, against all security threats identified in that application other than excessively costly brute

force attacks. (What constitutes excessive cost in this context is not clearly explained).

Again taking the February 1995 application's promises for context, "tamper resistance" requires that some barrier is in place which prevents access to or alteration of information in an unauthorized manner. The terms "secure" and "security", and additional terms such as "secure container," "control," "govern," "protect," "protected processing environment," "host processing environment" and "virtual distribution environment," would be understood, to the extent possible, as set forth in Microsoft's PLR 4-2 Statement, as opposed to the definitions listed in InterTrust's PLR 4-2 Statement.

4. Professor Mitchell will explain the qualifications of a person of reasonable skill in the computer security field, including as of February 13, 1995, and explain how cited references (such as U.S. Patent 5,634,012 to Stefik et al., U.S. Patents 4,868,877 and 5,337,360 to Fischer, Choudhury et al.'s "Copyright Protection for Electronic Publishing over Computer Networks," U.S. Patent 4,658,093 to Hellman, and Mori et al.'s "Superdistribution: The Concept and Architecture" (Transactions of the IECE 1990)) would influence such a person's understanding of the InterTrust disclosure. He may also address the substance of additional references published or created before February 13, 1995, not cited in the InterTrust patents.

5. The specifications of the '721, '900, and '861 patents do not resolve any of these problems with the Big Book application.



**Summary of Opinions of Professor David Maier  
in Support of Microsoft's Proposed Claim Constructions**

1. The specification of U.S. Patent No. 6,253,193 ("the '193 patent") describes several mandatory features of the Virtual Distribution Environment ("VDE") architecture, including:

- the creation of a comprehensive data security and commerce world;
- the ability to handle all types of digital works independent of computing platform, making it a single, general purpose solution in contrast to multiple, limited purpose solutions;
- flexible control mechanisms that can be applied to any granularity of content;
- control mechanisms that are configurable by any user, not just the system designers or content providers; and
- isolation of the system programs and protected works from the non-VDE world, preventing observation, alteration, interference, or removal from the VDE, except as permitted by the VDE control mechanisms.

This does not mean that the capabilities of the Virtual Distribution Environment can be achieved, only that these are features that the '193 patent makes clear a VDE must have.

2. The specification of the '193 patent describes a system that requires several architectural elements including at least the following:

- VDE Foundation Hardware and Software – installed throughout an infrastructure of interlinked computing devices;
- The VDE "Secure Container" – a mechanism for packaging protected works, control information, and administrative information; and

- The VDE “Control” – a mechanism for defining the regimen for using protected information that is inside a secure container.
3. Professor Maier will describe the background of a person of ordinary skill in the art. Such a person would understand the claims in light of the required capabilities and architectural features above.
4. The specification set forth in the ‘193 patent has numerous inconsistencies in its terminology. Some inconsistencies concern the data hierarchy (e.g., methods, control information, component assemblies). Other examples include the description of a non-secure host event processing environment and the concept of containment.

The following further summarizes Professor Maier’s opinions.

**I. EXPLANATION OF U.S. PATENT NO. 6,253,193**

**A. Asserted Capabilities of the Virtual Distribution Environment**

The ‘193 Patent describes a system that is asserted to be the first universal, distributed processing system for persistently controlling digital information. This system was given the name “Virtual Distribution Environment” or “VDE”. As described in the Patent, VDE promised at least the following mandatory features:

1. the creation of a comprehensive data security and commerce world;
2. the ability to handle all types of digital works independent of computing platform, making it a single, general purpose solution in contrast to multiple, limited solutions;
3. flexible control mechanisms that can be applied to any granularity of content;

4. control mechanisms that are configurable by any user, not just the system designers or content providers; and

5. isolation of the system programs and protected works from the non-VDE world, preventing observation, interference, or removal from the VDE, except as permitted by the VDE control mechanisms.

Although these features are promised by the '193 Patent, this does not mean that they are necessarily achievable.

#### **1. Comprehensive Data Security and Commerce World**

According to the '193 Patent, VDE is described as being the only comprehensive solution in a world of limited solutions. VDE is described as an end-to-end solution for digital works that guarantees the authenticity, confidentiality and integrity of the works and the VDE mechanisms. These protections are promised to be effective against any unauthorized activity by a third party (i.e. a user other than the creator of the work) that has physical possession of the computing hardware and wishes to circumvent the protections.

VDE must provide the ability to control the distribution and usage of digital works as well as tracking, reporting, auditing and handling payment for the distribution and usage. Additionally, VDE must support multiple business models simultaneously, for example, time-based and volume-based charging for the same digital work or licensing digital works with or without added sub-licensing rights.

Only those systems that are members of the electronic commerce world can participate in VDE commerce transactions. Consequently, all transactions must occur between

member systems since there is no way to control digital works that are outside the boundaries of the VDE world.

## **2. General Purpose**

According to the '193 Patent, the VDE system is the only rights management solution needed by its users because it is capable of handling and protecting all types of digital works, such as digital audio, digital video, software, digital cash, digital documents, electronic publications, etc. within a single rights management framework, whereas previous systems handled only limited subsets of information types. It further states that VDE can function within all types of electronic devices, from smart cards, pagers and telephones to supercomputers.

## **3. Flexible**

According to the '193 Patent, the VDE system can manage protected works in arbitrarily sized data chunks, down to the smallest atomic element. The Patent distinguished prior art systems that used access controls that were limited to the file level or resource level. The VDE system is described as being able to meter, track, bill and audit the usage of these arbitrary data chunks in addition to controlling the access to those data chunks. For example, a consumer can be charged by the number of bytes downloaded or by the number of paragraphs printed. Additionally, each of these actions can be specified independently, such that two objects can be metered differently, but billed identically.

This flexibility allows two different users to be charged at different rates, for different granularities, and in different currencies for using the same digital work. The '193 Patent distinguished prior art systems that lacked this flexibility.

#### **4. Controls Configurable by All Users**

According to the '193 Patent, the VDE system protects a digital work from the instant it is placed under VDE control subject to the permissions provided by the object creator (or rights holder) at the same or at another VDE "secure node." (The nature of the "secure node" is discussed later.) From that moment, the digital work becomes encapsulated within a VDE container. Then, the creator must grant permissions for accessing and distributing the digital work within the VDE object as well as identify how the object can be handled by other users of the VDE world.

These other users can create additional VDE-based controls for this protected work. In general, these controls only impose additional restrictions on the VDE object because they cannot conflict with the creator's VDE controls (except in the limited case in which the creator allows his controls to be modified by other users.) Even the end user is permitted to add VDE controls to VDE objects that he has received.

---

VDE controls are said to be persistent in that become permanently associated with the protected work once they are received, and they cannot be removed or deleted except as permitted by so-called "senior" VDE controls.

#### **5. System Isolation**

According to the '193 Patent, VDE protected works can only be accessed using VDE-certified foundation hardware and software. As a fundamental requirement, the VDE

foundation must isolate the internal workings of the system from the user because the user is not trusted.

Each computing device in the VDE world constitutes a "secure node" that must provide a "protected processing environment" (PPE) composed of VDE-certified foundation hardware and software. Sensitive materials such as protected works, administrative information, control information, and VDE software components, are passed between the protected processing environments of secure nodes inside "secure containers" that shield the materials from outside observation and alteration while in transit or in storage. The PPE must also shield all processing of the materials inside the PPE and also prevent the materials or process state information from "leaving" the VDE except as authorized by VDE control information. If the system fails to keep a protected work secret, then it can be distributed freely from that point onward. If the system fails to prevent alteration, then the consumer may receive invalid information (e.g., a bad stock quote), the consumer may receive less value than that for which he bargained (e.g., digital cash token that has been devalued), or the consumer's computer may be damaged by malicious code (e.g., virus-infected software), just to name a few examples. If the system fails to prevent the materials or process state information from leaving, then it can be moved to a system outside the VDE control regime for examination, manipulation, replication, or analysis.

Electronic devices outside the VDE world do not incorporate the VDE foundation, and hence are not constrained by VDE protocols. Thus, protected works are not permitted to be in clear text form outside of the isolated and rigidly controlled protected processing environment.

To guarantee the isolation and integrity of the PPE, the VDE foundation software itself must be protected by storing it in a location that is inaccessible to the user or by encrypting it when it is stored at a location that can be observed by the user.

#### **B. VDE Core Architecture**

According to the '193 Patent, three constituent building blocks are necessary to implement the VDE world:

1. VDE Foundation Hardware and Software – installed throughout an infrastructure of interlinked computing devices, each of which is called a “secure node”;
2. The VDE “Secure Container” – a mechanism for packaging protected works, control information, and administrative information; and
3. The VDE “Control” – a mechanism for defining the regimen for using protected information that is inside a secure container.

Both controls and protected works are transferred between secure nodes by means of the secure container mechanism. Secure containers can be opened (and the protected works used) only within the protected processing environment of a secure node by executing VDE controls that regulate and track such activity.

The proper combination of these three building blocks isolates internal processing from the untrusted user (by creating an unbyassable foundation of hardware and software); isolates protected works from the untrusted user (by placing them in a shielded data structure); and provides a control mechanism that will allow the untrusted user to make use of the protected works only under controlled conditions.

## **1. VDE Foundation Hardware/Software**

The VDE foundation hardware and software must ensure that the competing interests of both the owner and user of protected works are respected. The owner has an interest in controlling the distribution of his digital works and in compelling the reporting and payment for such use. The user has an interest in the control of his computing device, his privacy, and the availability of digital works for which he has paid.

The VDE foundation hardware and software must provide a sequestered venue in which external authority dominates the user's local authority in the control of information and processing. This VDE foundation hardware and software is the basis for any VDE installation on a device

A VDE secure node is a device that provides a VDE installation incorporating VDE foundation hardware and software as the base stratum on which all VDE functions are executed. In any secure node where protected works are used or where VDE control information is created or modified, a VDE secure subsystem core must be present. This core is enclosed by a "tamper resistant security barrier" that prevents observation of, interference with, and leaving of information and processes except as authorized by VDE control information.

This VDE secure subsystem core handles encrypting and decrypting data and code, storing control and metering information, managing secure communication with other VDE secure subsystem cores at other secure nodes, dynamically assembling and executing VDE control procedures, and updating control information for protected works.



Control procedures for the promised permission checking, metering, billing, and budget management features all execute within the VDE secure subsystem core.

The VDE foundation hardware and software must guarantee that control procedures triggered by user or system events are executed correctly and completely in the VDE secure subsystem core. Both correctness and completeness are necessary to preserve the integrity of VDE control regime. Failure can compromise the rights, privacy, or financial interests of the owner or user of the protected works.

According to the '193 Patent, these functions are provided and enforced by a secure processing unit (SPU) that is protected by a special purpose physical enclosure (the tamper resistant security barrier) that conceals the underlying VDE processing from observation or interference by external persons or processes, and that destroys information rather than allow the information to leave the VDE subsystem core via unauthorized means.

The '193 Patent suggests that a tamper resistant security barrier might be simulated solely in software by using several known software techniques, but it gives no specific direction as to how these techniques can be applied to achieve the guarantees required by the VDE secure subsystem core in an environment that is under the control of an untrusted user.

## **2. VDE Secure Containers**

An invariant requirement of the VDE container concept is that no access or use can be made of the protected works within a VDE container except as regulated by associated VDE control information. This associated control information can be provided in the

same secure container that holds the protected works or it can be provided independently in a separate secure container.

In addition to the protected works included within a secure container, there can be references to other digital works stored external to the container. However, the container cannot regulate other access or usage to these externally stored digital works. ("Containment" is discussed further in Section IV. D.)

VDE secure containers can contain administrative information, such as auditing, tracking, and billing requests and reports.

The internal structure of a VDE secure container must be able to store independently manageable digital works. Subsections of a VDE secure container can be encrypted by different keys, including subdivisions of a single digital work.

The internal structure of a VDE secure container must be able to store other VDE secure containers nested inside it. Each nested container is subject to its own independent control information. Control information corresponding to the outer container may not override more restrictive control information that corresponds to a secure container nested within it.

---

The VDE secure container supports modification of its contents and its control information subject to the current corresponding control information.

Because of this capability, a VDE secure container may be empty in the sense that it does not contain a digital work while it does contain control information that identifies the digital work that can be added to the secure container. Thus, a VDE secure container can be used as a mobile agent to retrieve digital works from remote locations.

### 3. VDE Controls

According to the '193 Patent, the configurability and flexibility of the VDE system arises jointly from the modular and independently selectable nature of control information and the dynamic construction and execution of control procedures within the VDE secure subsystem of a computing device. As used herein, the VDE secure subsystem refers to the VDE foundation hardware and software residing within the tamper resistant security barrier.

VDE controls are executable procedures constructed by the VDE foundation as a response to a request to access or use a specific protected work. The control is constructed inside the VDE secure subsystem using VDE control information. VDE control information is composed of executable code, rule information that is enforced by the executable code, and blueprint instructions for constructing the executable control. The VDE secure subsystem guarantees that the control procedure is constructed according to the blueprint instructions and that the components used in the construction are authentic as to source, identity, and data integrity.

All use of protected works is regulated by corresponding control information that is used to construct each executable control procedure. Different control procedures can regulate auditing, billing, metering, tracking and usage events (such as printing, rendering, copying, etc.) with respect to individual users for a single instance of a protected work. These events cannot occur except as regulated by the execution of the individual control procedures. Additionally, these control procedures can be applied at arbitrarily fine levels of granularity, such as charging for the number of bytes read.

Any VDE user can define control procedures to the extent permitted by senior VDE control information.

Control information is deliverable independent of the protected work. Individual portions of control information are deliverable independent of each other. Control information made by added, modified, or replaced over time to the extent permitted by earlier control information. Because independent control information for any given instance of a protected work can be created by different sources at different locations and different times, the control information from these sources can be in conflict. VDE must supply a means for resolving these conflicts. According to the '193 Patent, the executable controls negotiate to determine the conditions under which a protected work may be used. Thus, controls are said to "evolve" over time.

Once delivered to a VDE node with the corresponding protected work, control information persists throughout the life of the protected work.

The VDE controls must support a broad range of control regimes, all of which can co-exist on a single VDE secure node.

Dynamic assembly and execution of a VDE control must occur within the VDE secure subsystem. Construction of a VDE control from its component parts in a non-VDE system allows unconstrained access to digital works. Thus, VDE control information is transmitted between secure nodes using VDE secure containers and stored at VDE nodes in encrypted form whenever outside the VDE secure subsystem.

Executable control procedures are constructed from load modules, data, and VDE methods. These control procedures are assembled and executed in response to user and

system events. According to some statements in the '193 Patent, a "component assembly" is a VDE control procedure.

**C. Claim Interpretation**

A person of ordinary skill in the art would understand the claims of the '193 Patent in light of the mandatory capabilities and architectural components described above.

**D. Summary of Internal Inconsistencies.**

The '193 Patent contains numerous internal inconsistencies. Examples of these inconsistencies are given below.

**1. Use of Quotations**

Hundreds of terms are set off in quotations throughout the specification. These terms include: detail description, virtual distribution environment, electronic highway, VDE aware, content, virtual, things, chain of handling and control, rules and controls, CD ROM, information utility, switch, transaction processor, usage analyst, operating system, method, budget, atomic, firmware, hash bucket, peripheral device, event-based, multi-threaded, locking, Remote Procedure Call, two-phase commit, and read only. Some of these terms are coined (such as VDE aware; rules and controls; and usage analyst) while many are well known computer concepts (such as operating system and Remote Procedure Call.).

In many cases, it is unclear whether any particular use of quotation marks was intended to introduce a coined term, to indicate figurative or metaphorical usage of a term, to indicate non-standard or a weakened usage of a term, or something else

## 2. System Availability

In the Abstract, the '193 Patent asserts that "the invention . . . maintain[s] the integrity, availability, and/or confidentiality" of protected works. However, the system described does not appear to be designed to guarantee the availability of protected works. Rather, any deviation from the expected processing sequence is considered to be evidence of an attempt to crack the system or steal the protected works. In response, the system is likely to halt all processing until a trusted VDE administrator intervenes and resets the system. Additionally, the '193 Patent uses denial of service to enforce reporting obligations imposed by a rights holder. This practice is not consistent with preserving availability of digital works.

## 3. "Container" vs. "Object"

There is no consistent delineation in the '193 Patent between the terms "container" and "object." Initially, there appears to be a distinction in that the container is a shell data structure that is encapsulating data and the object is the combination of the container data structure and the encapsulated data. See Fig. 5A. Elsewhere, this distinction is blurred by the use of such phrases as:

"secure object (content container)";

"VDE content container is an object"; and

"VDE container (object)",

which appear to make container and object synonymous.

#### **4. The Property of Being "Contained"**

In the '193 Patent, there is no clear definition for the term "contain." The '193 patent states at one point that a container such as "container 302 may 'contain' items without those items actually being stored in the container." This definition of "contain" to include "referencing" is not customary in information storage terminology.

Subsequent examples in the '193 indicate that "contain" and "reference" are distinct relationships. For example, "may contain or reference" is used numerous times such as in "Load modules 1100 may contain or reference other load modules." and as in "Container 300y may contain and/or reference. . . ."

#### **5. Inconsistent Data Structure Hierarchy**

The hierarchy and relationships amongst rules, controls, methods, load modules, control information, and other data structures is inconsistent.

##### **a) "Rules and Controls" vs. "Control Information"**

The term "control information" is defined in the "Background and Summary of the Invention" of the '193 Patent as: ". . . load modules, associated data and methods . . ."

Later, the specification uses the phrase "'rules and controls' (control information)" as if the phrases "control information" and "rules and controls" are synonymous. Further, it states that "rules and controls" can be in the form of: "a 'permissions record' 808; 'budgets' 308 and 'other methods' 1000", but makes no mention of load modules.

Subsequent uses of "control information" such as: ". . . other aspects of the information to be contained within the object (e.g., rules and control information, identifying

information, etc.)"; and "the user may specify permissions, rules and/or control information." indicate that rules are different and distinct from control information.

**b) "Component Assembly" vs. "Control Information"**

In the '193 Patent, the relationship between component assembly and control information in the data hierarchy is defined inconsistently. Contrast the statement:

"In this example control information may include one or more component assemblies that describe the articles within such a container (e.g. one or more event methods referencing map tables and/or algorithms that describe the extent of each article)."

with:

*"... control information (typically a collection of methods related to one another by one or more permissions records, including any method defining variables) ..."*

[italics in original]

"This "channel 0" "open channel" task may then issue a series of requests to secure database manager 566 to obtain the "blueprint" for constructing one or more component assemblies 690 to be associated with channel 594 (block 1127). In the preferred embodiment, this "blueprint" may comprise a PERC 808 and/or URT 464."

In one case, the component assembly is a part of control information, but in the other case, control information is separable from and describes how to build a component assemblies.



c) **“Budgets”**

According to the '193 Patent, “budgets” are a special type of “method”. Methods are defined as containing, among other things, “User Data Elements”. Elsewhere, budgets are cited as a common type of User Data Element. This inconsistency creates confusion as to whether any given use of the term “budget” refers to an executable method or a non-executable data structure.

6. **“Load Module”**

According to the '193 Patent, executable code is provided in the form of “atomic” load modules”, presumably meaning that they are the smallest unit of executable code. Later, however, load modules are sub-dividable into smaller load modules, which is inconsistent with atomicity.

7. **The “Non-Secure” “Protected Processing Environment”**

According to the '193 Patent, a necessary feature of a VDE computer is the “protected processing environment” or “PPE”. Secure Event Processing Environments (“SPE”), in which all sensitive processing is handled inside a hardware device called a Secure Processing Unit (“SPU”) are stated as being one type of PPE. Host Event Processing Environments (“HPE”) are also said to be a type of PPE. The HPE classification is further described as having two sub-types: secure and non-secure. Additionally, the specification defines the three abbreviations as synonymous and interchangeable starting at column 103 of the specification, unless the context of any given passage indicates otherwise.

Further, no criteria are provided for distinguishing between a "secure HPE" and a "non-secure HPE". Thus, it is not possible to reconcile the "non-secure HPE" as a secure operating environment or protected processing environment.

## EXHIBIT H

### Mini Markman 30 Terms/Phrases to Address

1. Set forth below are the twelve claims designated for the "Mini-Markman" proceeding.
2. The parties, in accordance with the Court's February 21, 2003, Order, have agreed to narrow the "Mini-Markman" proceeding to a selected thirty terms and phrases, set forth in boldface below.
3. Bold denotes the terms and the phrases that the parties have designated to be construed in the "Mini-Markman" proceeding; underscoring denotes the designation is a phrase.
4. Bolding of the claim number indicates that Microsoft construes the claim as a whole as requiring its "Global Construction" of "VDE."

### U.S. Patent No. 6,253,193

1. A method comprising:  
receiving a digital file including music;  
storing said digital file in a first **secure** memory of a first device;  
storing information associated with said digital file in a **secure** database stored on said first device,  
said information including at least one **budget control** and at least one **copy control**, said at  
least one **budget control** including a budget specifying the number of copies which can be  
made of said digital file; and said at least one copy control controlling the copies made of  
said digital file;  
determining whether said digital file may be **copied** and stored on a second device based on at least  
said **copy control**;  
if said **copy control** allows at least a portion of said digital file to be **copied** and stored on a second  
device,  
copying at least a portion of said digital file;  
transferring at least a portion of said digital file to a second device including a memory and an  
audio and/or video output;  
storing said digital file in said memory of said second device; and  
including playing said music through said audio output.

11. A method comprising:  
receiving a digital file;  
storing said digital file in a first **secure** memory of a first device;  
storing information associated with said digital file in a **secure** database stored on said first device,  
said information including a **first control**;  
determining whether said digital file may be **copied** and stored on a second device based on said **first  
control**,  
said determining step including identifying said second device and determining whether said  
**first control** allows transfer of said **copied** file to said second device,

said determination based at least in part on the features present at the device to which said **copied** file is to be transferred;  
if said first **control** allows at least a portion of said digital file to be **copied** and stored on a second device,  
    **copying** at least a portion of said digital file;  
    transferring at least a portion of said digital file to a  
        second device including a memory and an audio and/or video output;  
    storing said digital file in said memory of said second device; and  
    rendering said digital file through said output.

**15. A method comprising:**

receiving a digital file;

an **authentication** step comprising:

    accessing at least one **identifier** associated with a first device or with a user of said first device;  
    and

    determining whether said **identifier** is associated with a device and/or user authorized to store said digital file;

storing said digital file in a first **secure** memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized;

storing information associated with said digital file in a **secure** database stored on said first device, said information including at least one **control**;

determining whether said digital file may be **copied** and stored on a second device based on said at least one **control**;

if said at least one **control** allows at least a portion of said digital file to be **copied** and stored on a second device,

**copying** at least a portion of said digital file;

    transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;

    storing said digital file in said memory of said second device; and  
    rendering said digital file through said output.

**19. A method comprising:**

receiving a digital file at a first device;

establishing communication between said first device and a **clearinghouse** located at a location remote from said first device;

    said first device obtaining authorization information including a key from said **clearinghouse**;

    said first device using said authorization information to gain access to or make at least one **use** of said first digital file, including using said key to decrypt at least a portion of said first digital file; and

    receiving a first **control** from said **clearinghouse** at said first device;

storing said first digital file in a memory of said first device;

using said first **control** to determine whether said first digital file may be **copied** and stored on a second device;  
if said first **control** allows at least a portion of said first digital file to be **copied** and stored on a second device,  
**copying** at least a portion of said first digital file;  
transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;  
storing said first digital file portion in said memory of said second device; and  
rendering said first digital file portion through said output.

U.S. Patent No. 6,185,683

2. A system including:  
a first apparatus including,  
    user **controls**,  
    a communications port,  
    a processor,  
    a memory storing:  
        a first **secure container** containing a governed item, the first **secure container** governed item being at least in part encrypted; the first **secure container** having been received from a second apparatus;  
        a first **secure container** rule at least in part governing an **aspect** of access to or use of said first **secure container** governed item, the first **secure container** rule, the first **secure container** rule having been received from a third apparatus different from said second apparatus; and  
        hardware or software used for receiving and opening **secure containers**, said **secure containers** each including the capacity to **contain** a governed item, a **secure container** rule being associated with each of said **secure containers**;  
a **protected processing environment** at least in part protecting information **contained** in said **protected processing environment** from tampering by a user of said first apparatus, said **protected processing environment** including hardware or software used for applying said first **secure container** rule and a second **secure container** rule in combination to at least in part govern at least one **aspect** of access to or use of a governed item **contained** in a **secure container**; and  
hardware or software used for transmission of **secure containers** to other apparatuses or for the receipt of **secure containers** from other apparatuses.

U.S. Patent No. 6,157,721

1. A security method comprising:  
(a) **digitally signing** a first load module with a first **digital signature** designating the first load module for use by a first device class;

- (b) digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class;
- (c) distributing the first load module for use by at least one device in the first device class; and
- (d) distributing the second load module for use by at least one device in the second device class.

34. A protected processing environment comprising:

a first tamper resistant barrier having a first security level,  
a first secure execution space, and

at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.

**U.S. Patent No. 5,920,861**

58. A method of creating a first secure container, said method including the following steps;  
accessing a descriptive data structure, said descriptive data structure including or addressing organization information at least in part describing a required or desired organization of a content section of said first secure container, and  
metadata information at least in part specifying at least one step required or desired in creation of said first secure container;  
using said descriptive data structure to organize said first secure container contents;  
using said metadata information to at least in part determine specific information required to be included in said first secure container contents; and  
generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents.

**U.S. Patent No. 5,982,891**

1. A method for using at least one resource processed in a secure operating environment at a first appliance, said method comprising:  
securely receiving a first entity's control at said first appliance, said first entity being located remotely from said operating environment and said first appliance;  
securely receiving a second entity's control at said first appliance, said second entity being located remotely from said operating environment and said first appliance, said second entity being different from said first entity; and  
securely processing a data item at said first appliance, using at least one resource, including securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item.

U.S. Patent No. 5,892,900

155. A virtual distribution environment comprising  
a first host processing environment comprising  
a central processing unit;  
main memory operatively connected to said central processing unit;  
mass storage operatively connected to said central processing unit and said main memory;  
said mass storage storing tamper resistant software designed to be loaded into said main memory and  
executed by said central processing unit, said tamper resistant software comprising:  
machine check programming which derives information from one or more aspects of said host  
processing environment,  
one or more storage locations storing said information;  
integrity programming which causes said machine check programming to **derive** said information,  
**compares** said information to information previously stored in said one or more storage  
locations, and  
generates an indication based on the result of said **comparison**; and  
programming which takes one or more actions based on the state of said indication;  
said one or more actions including at least temporarily halting further processing.

U.S. Patent No. 5,917,912

8. A process comprising the following steps:  
accessing a first record **containing** information directly or indirectly identifying one or more elements  
of a first **component assembly**,  
at least one of said elements including at least some **executable programming**,  
at least one of said elements constituting a load module,  
said load module including **executable programming** and a header;  
said header including an execution space **identifier identifying at least one aspect of an  
execution space required for use and/or execution of the load module** associated  
with said header;  
said execution space **identifier** provides the capability for distinguishing between  
execution spaces providing a higher level of security and execution spaces  
providing a lower level of security;  
using said information to identify and locate said one or more elements;  
accessing said located one or more elements;  
**securely** assembling said one or more elements to form at least a portion of said first **component  
assembly**;  
executing at least some of said **executable programming**; and  
checking said record for validity prior to performing said executing step.

35. A process comprising the following steps:  
at a first processing environment receiving a first record from a second processing environment remote  
from said first processing environment;

said first record being received in a **secure container**;  
said first record **containing** identification information directly or indirectly identifying one or more elements of a **first component assembly**;  
at least one of said elements including at least some **executable programming**;  
said **component assembly** allowing access to or use of specified information;  
said **secure container** also including a first of said elements;  
accessing said first record;  
using said identification information to identify and locate said one or more elements;  
said locating step including locating a second of said elements at a third processing environment located remotely from said first processing environment and said second processing environment;  
accessing said located one or more elements;  
said element accessing step including retrieving said second element from said third processing environment;  
**securely** assembling said one or more elements to form at least a portion of said **first component assembly** specified by said first record; and  
executing at least some of said **executable programming**,  
said executing step taking place at said first processing environment.



## EXHIBIT I

### PLR 4-3(a) – Constructions on Which the Parties Agree

|     | Claim Term / Phrase  | Agreed Construction  |
|-----|--|--|
| 1.  | <b>entity</b><br>891.1   | Any person or organization.  |
| 2.  | <b>generating</b><br>861.58  | Producing.   |
| 3.  | <b>govern, governed, governing</b><br>891.1, 683.2   | See Control (v.).  |
| 4.  | <b>metadata information</b><br>861.58  | Information that describes one or more attributes of other data, and/or the processes used to create and/or use that data. For example, metadata information may describe the following attributes of other data: its meaning, representation in storage, what it is used for and by whom, context, quality and condition, location, ownership, or its data elements or their attributes (name, size, data type, etc.) |
| 5.  | <b>rendering</b><br>193.11, 193.15, 193.19   | In the context of 193.11, 15 and 19: Playing content through an audio output (e.g., speakers) or displaying content on a video output (e.g., a screen).  |
| 6.  | <b>secure container rule</b><br>683.2  | A “rule” that governs (Controls) a Secure Container “governed item.”   |
| 7.  | <b>security</b><br>721.1, 721.34   | See Secure.  |
| 8.  | <b>tampering</b><br>683.2, 721.1, 721.34, 900.155  | Using (e.g., observing or altering) in any unauthorized manner, or interfering with authorized use.  |
| 9.  | <b>“said mass storage storing tamper resistant software”</b><br>900.155                              | The “tamper resistant software” is physically stored within, as opposed to being merely “addressed” by, the mass storage.  |
| 10. | <b>“including using said key to decrypt at least a portion of said first digital file”</b><br>193.19 | The “at least one use of said digital file” must encompass decrypting at least a “portion” of the “digital file” using the “key.”  |

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**